Journal Subline

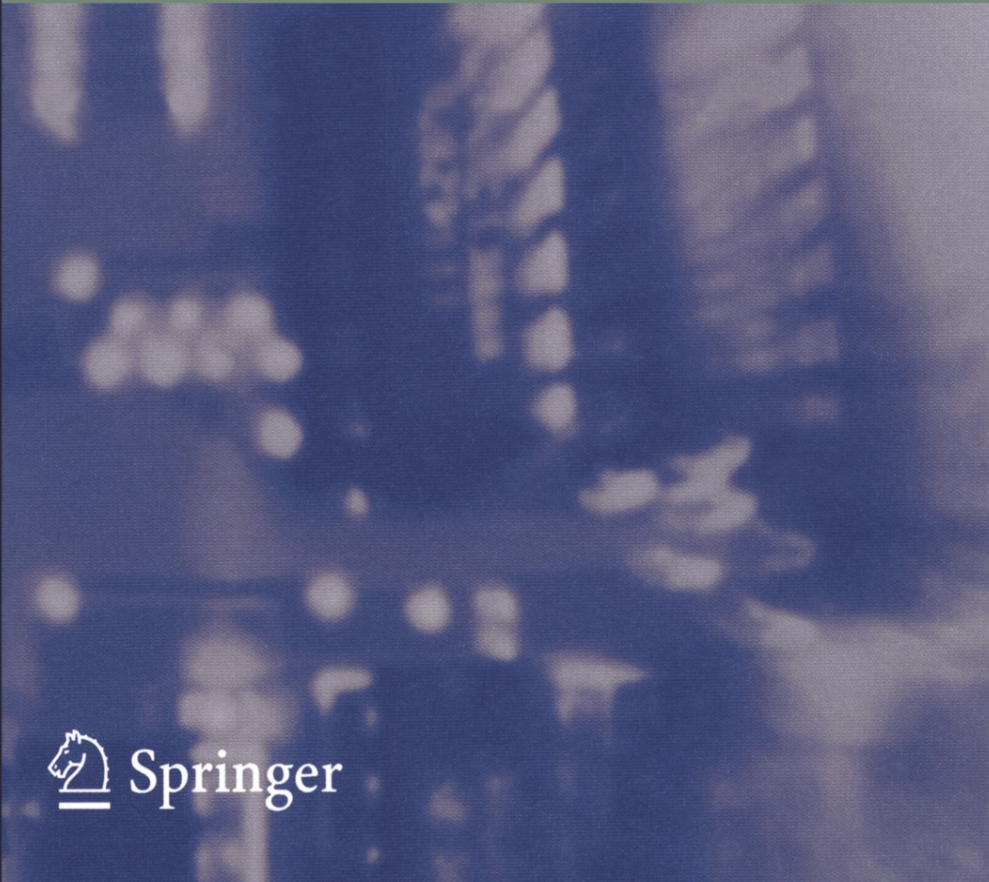
LNCS 3135

Transactions on Rough Sets II

Rough Sets and Fuzzy Sets

James F. Peters Andrzej Skowron

Editors-in-Chief



Springer

Lecture Notes in Computer Science 3135

James F. Peters Andrzej Skowron
Didier Dubois Jerzy W. Grzymała-Busse
Masahiro Inuiguchi Lech Polkowski (Eds.)

Transactions on Rough Sets II

Rough Sets and Fuzzy Sets

Springer

Volume Editors

Didier Dubois
Université Paul Sabatier, CNRS, IRIT
118, route de Narbonne, 31062 Toulouse Cedex 4, France
E-mail: dubois@irit.fr

Jerzy W. Grzymała-Busse
University of Kansas, Department of Electrical Engineering and Computer Science
3014 Eaton Hall, 1520 W. 15th St., #2001, Lawrence, KS 66045-7621, USA
E-mail: jerzy@ku.edu

Masahiro Inuiguchi
Osaka University, Department of Mathematical Science for Social Systems
Department of Systems Innovation, Graduate School of Engineering Science
Machikaneyama-cho 1-3, Toyonaka, Osaka 560-8531, Japan
E-mail: inuiguti@sys.es.osaka-u.ac.jp

Lech Polkowski
Polish–Japanese Institute of Information Technology
and Department of Mathematics and Computer Science
University of Warmia and Mazury
Koszykowa 86, 02-008 Warsaw, Poland
E-mail: polkow@pjwstk.edu.pl

Library of Congress Control Number: 2004115993

CR Subject Classification (1998): F.4.1, F.1, I.2, H.2.8, I.5.1, I.4

ISSN 0302-9743
ISBN 3-540-23990-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11322016 06/3142 5 4 3 2 1 0

LNCS Transactions on Rough Sets

This journal subline has as its principal aim the fostering of professional exchanges between scientists and practitioners who are interested in the foundations and applications of rough sets. Topics include foundations and applications of rough sets as well as foundations and applications of hybrid methods combining rough sets with other approaches important for the development of intelligent systems.

The journal includes high-quality research articles accepted for publication on the basis of thorough peer reviews. Dissertations and monographs up to 250 pages that include new research results can also be considered as regular papers. Extended and revised versions of selected papers from conferences can also be included in regular or special issues of the journal.

Editorial Board

Preface

This collection of articles is devoted to fuzzy as well as rough set theories. Both theories are based on rigorous ideas, methods and techniques in logic, mathematics, and computer science for treating problems for which approximate solutions are possible only, due to their inherent ambiguity, vagueness, incompleteness, etc. Vast areas of decision making, data mining, knowledge discovery in data, approximate reasoning, etc., are successfully explored using methods worked out within fuzzy and rough paradigms.

By the very nature of fuzzy and rough paradigms, outlined above, they are related to distinct logical schemes: it is well-known that rough sets are related to modal logics $S5$ and $S4$ (Orłowska, E., Modal logics in the theory of information systems, *Z. Math. Logik Grund. Math.* 30, 1984, pp. 213 ff.; Vakarelov, D., Modal logics for knowledge representation systems, *LNCS* 363, 1989, pp. 257 ff.) and to finitely-valued logics (Pagliani, P., Rough set theory and logic-algebraic structures. In *Incomplete Information: Rough Set Analysis*, Orłowska, E., ed., Physica/Springer, 1998, pp. 109 ff.; Polkowski, L. A note on 3-valued rough logic accepting decision rules, *Fundamenta Informaticae* 61, to appear).

Fuzzy sets are related to infinitely-valued logics (fuzzy membership to degree $r \in [0,1]$ expressing truth degree r) (Goguen, J.A., The logic of inexact concepts, *Synthese* 18/19, 1968–9, pp. 325 ff.; Pavelka, J., On fuzzy logic I, II, III, *Z. Math. Logik Grund. Math.* 25, 1979, pp. 45 ff., pp. 119 ff., pp. 454 ff.; Dubois, D., Prade, H., *Possibility Theory*, Plenum Press, 1988; Hájek, P., *Metamathematics of Fuzzy Logic*, Kluwer, 1998).

Algebraic as well as topological features of roughness and fuzziness are distinct. Topologically, rough sets may be described by means of topologies on families of sets (Polkowski, L., *Rough Sets. Mathematical Foundations*, Physica/Springer, 2002) whereas fuzzy sets by their nature fall into the province of topologies on function spaces (Ying-Ming Liu, Mao Kang Luo, *Fuzzy Topology*, World Scientific, 1998). Algebraically, rough sets form structures known as Łukasiewicz algebras, Heyting algebras, Post algebras, etc. (Pagliani, op. cit., Polkowski, op. cit.), whereas fuzzy set algebra involves point-wise operations on fuzzy membership functions suggested by various logical interpretations of fuzzy union, intersection, etc. (Novák, V., Perfilieva, I., Močkoř, J., *Mathematical Principles of Fuzzy Logic*, Kluwer, 1999).

Despite some differences, there have been attempts to reconcile the two theories and to form a hybrid paradigm, rough-fuzzy, or fuzzy-rough, depending on whether rough constructs are introduced in the fuzzy set framework, or conversely, fuzzy constructs are defined in the rough set framework (Dubois, D., Prade, H., Putting rough sets and fuzzy sets together. In *Intelligent Decision Systems. Handbook of Applications and Advances of Rough Sets Theory*, Słowiński, R., ed., Kluwer, 1992, pp. 203 ff.; Dubois, D., Prade, H., Similarity versus preference in fuzzy-set based logics. In *Incomplete Information: Rough Set Analysis*, Orłowska, E., ed., Physica/Springer, 1998, pp. 441 ff.; Nakamura, A., Fuzzy rough sets, *Notes on Multiple-Valued Logic in Japan*, 9, 1988, pp. 1 ff.; Cattaneo, G., Generalized rough sets. Preclusivity fuzzy-intuitionistic (BZ) lattices, *Studia Logica*, 58, 1997, pp. 47 ff.; Pedrycz, W., Shadowed sets: bridging fuzzy and rough sets. In *Rough Fuzzy Hybridization*, Pal, S.K., Skowron, A., eds., Springer, Singapore, 1999, pp. 179 ff.; Inuiguchi, M., Tanino, T., A new class of necessity measures and fuzzy rough sets based on certainty qualifications, *LNAI* 2005, 2000, pp. 261 ff.).

The volume presented to the reader contains papers devoted to rough set theory, to fuzzy set theory, and to both theories. These papers highlight important aspects of those theories from theoretical as well as application points of view.

It is our pleasure that this volume appears in the Lecture Notes in Computer Science series of Springer-Verlag in the newly initiated sub-series of Transactions on Rough Sets. We are indebted to the editors of the subseries, Profs. Peters and Skowron for their invitation to publish the volume in this subseries. Our thanks go also to Prof. Janusz Kacprzyk who suggested that we prepare a collection of papers devoted simultaneously to rough and fuzzy theories. We would like to thank the authors, whose chapters are included in this volume, for making this possible. Our thanks go to the editors of Springer-Verlag, for their dedicated work toward giving the volume its final shape as well as to Dr. Piotr Synak who helped us with LaTeX.

May 2004

Didier Dubois
Jerzy Grzymała-Busse
Masahiro Inuiguchi
Lech Polkowski

Table of Contents

Feature Selection with Rough Sets for Web Page Classification

Aijun An[1], Yanhui Huang[2], Xiangji Huang[2], and Nick Cercone[3]

[1] York University, Toronto, Ontario, M3J 1P3, Canada
[2] University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
[3] Dalhousie University, Halifax, Nova Scotia, B3H 1W5, Canada

Abstract. Web page classification is the problem of assigning predefined categories to web pages. A challenge in web page classification is how to deal with the high dimensionality of the feature space. We present a feature reduction method based on the rough set theory and investigate the effectiveness of the rough set feature selection method on web page classification. Our experiments indicate that rough set feature selection can improve the predictive performance when the original feature set for representing web pages is large.

1 Introduction

With the rapid growth of information on the World Wide Web, automatic classification of web pages has become important for effective indexing and retrieval of web documents. One approach to automatic web page classification is to apply machine learning techniques to pre–classified web data to induce profiles of categories and compare the profiles of categories with the representation of a given document in order to classify the document. A major characteristic, or difficulty, of this application is the high dimensionality of the feature space. A common approach to representing a text document is to use a "bag of words" that appear in the document. Since a web page can contain thousands of words, the feature space for representing web pages is potentially huge. Few machine learning systems can handle such a large number of features. In addition, too many features may present noise to the learning system. Therefore, it is highly desirable to reduce the feature space in order to make use of existing learning systems, to improve classification accuracy, and to speed up the learning process. It is also desirable to achieve such a goal automatically, i.e., no manual selection or construction of features is required.

Automatic feature selection methods have been used in text classification. Lewis and Ringuette [6] used an information gain measure to reduce the document vocabulary in naive Bayes classification and decision tree learning. Yang [15] used the principal component analysis to find orthogonal dimensions in the vector space of documents. Wiener *et al.* [14] used mutual information and a χ^2 statistic to select features for input to neural networks. Lang [4] used a minimum description length principle to select terms for news categorization. It has been asserted that feature selection is the most critical stage of the learning process in text classification [6].

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 1–13, 2004.

We investigate the effectiveness of feature selection by rough sets on web page classification. Rough set theory is a mathematical tool for modelling incomplete or imprecise information [9]. It has been used for both feature selection and knowledge discovery in a number of real world domains, including medicine, pharmacology, control systems, social sciences, switching circuits, and image processing [13], [11]. In this paper, we apply rough set theory to feature selection for web page classification. In our application, web pages in a training data set are first represented using top frequent words. Then a feature selection method based on rough sets is applied to remove redundant features from the training data. A rule induction system, named ELEM2 [1], is then used to learn classification rules from the reduced training data. Therefore, in our application, the rough sets based feature selection is used as a pre–processing step for ELEM2. To evaluate the effectiveness of rough set feature selection on web page classification, we conduct experiments to compare the predictive performances of ELEM2 on web page classification with and without rough set feature selection. We describe our experiments and report the evaluation results.

The chapter is organized as follows. In the next section, we describe the importance of web page classification and the problems that need to be solved for web page classification. We also present our data collection and representation methods. In Sect. 3, we present the basic concepts in rough sets and describe an algorithm for computing a *reduct*,i.e., a non–redundant subset of features. The ELEM2 rule induction method is briefly introduced in Sect. 4. Our method for classifying a web page is presented in Sect. 5. In Sect. 6, we describe our evaluation methods and report experimental results. Finally, we conclude the paper in Sect. 7.

2 The Problem of Web Page Classification

The World Wide Web contains an estimate of 968 million pages as of March 2002 in the Google search engine [8] and an estimate of 7 million or more pages being added daily [5]. Describing and organizing this vast amount of content is essential for realizing the web as an effective information resource. Text classification has become an important process for helping web search engines to organize this vast amount of data. For instance, most Internet search engines, such as Yahoo and Looksmart, divide the indexed web documents into a number of categories for the users to limit the search scope. Moreover, text classification makes the results easier to browse. When the results returned by the search engine have been classified into a specified category, the users can choose the interesting category to continue browsing. Traditionally, text classification is performed manually by domain experts. However, human classification is unlikely to keep pace with the rate of growth of the web. Hence, as the web continues to increase, the importance of automatic web page classification becomes obvious. In addition, automatic classification is much cheaper and faster than human classification.

To make the text classification process automatic, machine learning techniques can be applied to generate classification models from a set of text documents with pre-labelled categories. The classification model can then be used to automatically assign natural language texts to the predefined categories based on their contents. In order to apply a machine learning technique to web page classification, the following problems need to be solved. First, to build a web page classifier, we need to collect a set of web pages as training examples to train the machine learning system. These training examples should have pre-defined class labels. Second, the content of a web page in the training set should be analyzed and the page should be represented using a formalism that the learning system requires for representing training examples. This text representation issue is central to our application. Finally, how to classify new pages with induced rules is another challenge in our application. We use different sets of features to represent training pages of different categories. Therefore, the rules for different categories are expressed using different sets of features. When classifying a new page, these different sets of rules should be used together in some way to determine the category or categories of the new page.

2.1 Data Collection

We use the Yahoo web site to collect training examples for our learning problem. Yahoo is best known for maintaining a web categorization directory. The web directory in Yahoo is a multi–level tree–structured hierarchy. The top level of the tree, which is the first level below the root of the tree, contains 14 categories. Each of these 14 categories contains sub–categories that are placed in the second level below the root. The third and fourth levels of the tree contain both further–refined categories and web pages. We use the top–level categories in Yahoo to label the web pages in our training set. Only 13 of 14 top–level categories are used and one category, named “Regional”, is excluded because it has too much overlap with other categories.

We randomly selected over 7600 pages from the Yahoo category. We originally planned to gather 500 example pages from each category. Unfortunately, some web pages that have links in Yahoo were either eliminated or not connected to the Internet. In addition, some web pages contain very few terms after the removal of stop–words because these pages consist of a brief greeting sentence, image, Java script, flash, and other non–textual information. Hence, our number of training examples for each category is different. The distribution of these training examples among 13 categories considered is shown in Table 1. Categories considered may overlap. For example, a document discussing sport action may be reasonably classified into Entertainment and Recreation & Sports categories.

2.2 Representation of Web Pages

After training pages are selected, we apply Porter’s stemming algorithm [10] to transfer each word in a web page into its stem. We then remove all the stop words according to a standard stop word list. For each category, we count the number of occurrences of each remaining word stem in all the pages that belong to the category. The word stems in each category are then sorted according to

Table 1. Distribution of the training data.

Category	Number of web pages
Arts & Humanities	783
Business & Economy	997
Computers & Internet	745
Education	485
Entertainment	957
Government	229
Health	772
News & Media	747
Recreation & Sports	506
Reference	501
Society & Culture	253
Science	230
Social Science	510
Total	7615

the number of occurrences. This process results in two sets of documents. One is 13 sorted lists of word stems, one for each category. These lists will be used to select features for the training data. The other set of results is the set of web pages, each represented by remaining word stems and their counts.

We use word stem counts to represent a web document. A web document may contain a huge number of words and not all the words in the global space appear in every document. If we use all the words in the global space to represent the documents, the dimensionality of the data set is prohibitively high for the learning system. In addition, even though our learning system can handle thousands of features, many of the features are irrelevant to the learning task. The presence of irrelevant features in the training data introduces noise and extra learning time. Therefore, it is necessary to reduce the dimensionality of the feature set by removing words with low frequencies. Removing infrequent words is also suggested in [7] and [16].

Different categories have different top frequent words. We collected top 60 frequent terms for each category. Since the top frequent terms differ among categories, there is no common set of features that we can use to represent all documents in all categories. Therefore, even though our learning system can deal with multi–category learning directly, we transform our learning problem into multiple two–class learning problems. That is, for each web page category, we prepare the training data using top n ($n = 20, 30, 40, 50$ or 60 in our experiments) frequent words in the category and then learn a set of rules that can be used to predict whether a new page belongs to this category or not. Therefore, for a given n, we have 13 training sets , each of them containing 7,615 documents, and represented by top n frequent terms of the corresponding category. After applying our learning algorithm, the 13 training sets lead to the generation of 13 classifiers. These 13 classifiers will vote to determine which category or categories a new page belongs to.

3 Feature Selection with Rough Sets

We use frequent words to represent the web pages in our training data. However, some frequent words may not be very relevant to our learning task. These words may have little power in discriminating documents of different categories. Therefore, further selection of relevant features is important. We apply a rough set based feature selection method for this purpose. In this section, we first introduce some concepts of rough sets and then describe an algorithm for removing unnecessary attributes.

3.1 Basic Notations

A data set can be formally described using a *decision table.* A decision table (also called an information system [9]) is defined as a quadruple $< U, A, V, f >$, where $U = x_1, x_2, ..., x_N$ is a finite set of objects or examples; A is a finite set of attributes; the attributes in A are further classified into two disjoint subsets, *condition* attributes C and *decision* attributes D such that $A = C \cup D$ and $C \cap D = \emptyset$; $V = \bigcup_{a \in A} V_a$ is a set of attribute values and V_a is the *domain* of attribute a (the set of values of attribute a); $f : U \times A \to V$ is an *information function* which assigns particular values from domains of attributes to objects such that $f(x_i, a) \in V_a$, for all $x_i \in U$ and $a \in A$. In our application, $D = \{d\}$ is a singleton set, where d is the class attribute that denotes classes of examples.

Given a decision table $DT =< U, A, V, f >$, let B be a subset of A, and let x_i and x_j be members of U; a relation $R(B)$, called an *indiscernibility relation* [9] over B, is defined as follows:

$$R(B) = \{(x_i, x_j) \in U^2 : \forall a \in B, f(x_i, a) = f(x_j, a)\}. \tag{1}$$

Let C be the set of condition attributes, and $R(C)$ be the corresponding indiscernibility relation on U; an ordered pair $AS =< U, R(C) >$ is called an *approximation space* based on C.

Let $Y \subseteq U$ be a subset of objects representing a *concept*, and $R^*(C) = \{X_1, X_2, ..., X_n\}$ be the collection of equivalence classes induced by the relation $R(C)$. The *lower approximation* [1] of a set Y in the approximation space AS denoted as $LOW_{R(C)}(Y)$, is defined as the union of those equivalence classes in the collection of $R^*(C)$ which are completely contained in the set Y, i.e.,

$$LOW_{R(C)}(Y) = \bigcup \{X \in R^*(C) : X \subseteq Y\}. \tag{2}$$

Let $R^*(D) = \{Y_1, Y_2, ..., Y_m\}$ be the collection of equivalence classes of the relation $R(D)$. The *positive region* $POS_C(D)$ of D with respect to C is defined as,

$$POS_C(D) = \bigcup_{i=1,...,m} \{LOW_{R(C)}(Y_i) : Y_i \in R^*(D)\}. \tag{3}$$

The positive region $POS_C(D)$ includes all examples of the equivalence classes of $R^*(C)$ in AS which can be certainly classified into classes of $R^*(D)$. $POS_C(D) = U$ means that there is no conflict between sets C and D in the sense that the classification by C determines the classification by D on objects in U.

3.2 Attribute Reduction

Attribute reduction techniques eliminate superfluous attributes and create a minimal sufficient subset of attributes for a decision table. Such minimal sufficient subset of attributes, called a *reduct*, is an essential part of the decision table which can discern all examples discernible by the original table and cannot be reduced any more. A subset B of a set of attributes C is a *reduct* of C with respect to D if and only if,

(1) $POS_B(D) = POS_C(D)$,
(2) $POS_{B-\{a\}}(D) \neq POS_C(D)$, for any $a \in B$.

A set C of condition attributes may contain more than one reduct. The set of attributes common to all reducts of C is called the *core* of C. The core contains all indispensable attributes of a decision table and can be defined as,

$$CORE_C(D) = \{c \in C : \forall c \in C, POS_{C-\{c\}}(D) \neq POS_C(D)\}. \tag{4}$$

A good procedure for computing a reduct for a decision table is to compute the core first and then check the other attributes one by one to see if they are essential to the system. If for any attribute $c \in C - CORE_C(D)$, $POS_{C-\{c\}}(D) \neq POS_C(D)$, then c cannot be removed from C. Since the order in which the attributes are removed affects the result of reduction, a concept called *relative significance coefficient (RSC)* is introduced to rank the condition attributes. The *relative significance coefficient* (RSC) of the attribute $c \in C$ based on the set of attributes C with respect to attributes D is defined as,

$$RSC_c(C, D) = \frac{card(POS_{C-\{c\}}(D))}{card(POS_C(D))}, \tag{5}$$

where *card* denotes set cardinality. Our algorithm for computing a reduct is outlined as follows.

1. Compute $CORE_C(D)$. For each condition attribute in C, remove it from C and check whether it changes the positive region. Let $CORE_C(D)$ be the set of all condition attributes whose removal does not change the positive region.
2. Check whether $CORE_C(D)$ is a reduct of the rule set. If yes, stop and $CORE_C(D)$ is a reduct.
3. Let $T = C - CORE_C(D)$. Rank the attributes in T in descending order of their RSC value. Let a be the first attribute in T and let C' be C.
4. Check whether $POS_{C'-\{a\}}(D) = POS_C(D)$. If yes, remove a from C'.
5. Let a be the next attribute in T. If a exists, repeat step 4; otherwise, stop and C' is a reduct.

4 ELEM2 Rule Induction

ELEM2 [1] is a rule induction system that learns classification rules from a set of data. Given a set of training data, ELEM2 sequentially learns a set of rules for each class in the data set. To induce rules for a class C, ELEM2 conducts general–to–specific heuristic search over a hypothesis space to generate a disjunctive set of propositional rules. ELEM2 uses a *sequential covering* learning strategy; it reduces the problem of learning a disjunctive set of rules to a sequence of simpler problems, each requiring that a single conjunctive rule be learned that covers a subset of positive examples. The learning of a single conjunctive rule begins by considering the most general rule precondition, i.e., the empty test that matches every training example, then greedily searching for an attribute-value pair that is most relevant to the class C according to the following attribute–value pair evaluation function,

$$SIG_C(av) = P(av)(P(C|av) - P(C)),$$

where av is an attribute–value pair and P denotes probability. The selected attribute–value pair is then added to the rule precondition as a conjunct. The process is repeated by greedily adding a second attribute–value pair, and so on, until the hypothesis reaches an acceptable level of performance. In ELEM2, the acceptable level is based on the consistency of the rule: it forms a rule that is as consistent with the training data as possible. Since this "consistent" rule may be a small disjunct that overfits the training data, ELEM2 "post-prunes" the rule after the initial search for this rule is complete. To post-prune a rule, ELEM2 computes a rule quality value for the rule according to one of the rule quality formulas described in [2][1]. ELEM2 then checks each attribute–value pair in the rule in the order reverse to that in which they were selected, to determine whether removal of the attribute–value pair will decrease the rule quality value. If not, the attribute–value pair is removed and the procedure checks all the other pairs in the same order again using the new rule quality value resulting from removal of that attribute–value pair to determine whether another attribute–value pair can be removed. This procedure continues until no pair can be removed.

After rules are induced for all the classes, the rules can be used to classify new examples. The classification procedure in ELEM2 considers three possible cases when matching a new example with a set of rules.

1. *Single match.* The new example satisfies one or more rules of the same class. In this case, the example is classified to the class indicated by the rule(s).
2. *Multiple match.* The new example satisfies more than one rule that indicates different classes. In this case, ELEM2 activates a conflict resolution scheme for the best decision. The conflict resolution scheme computes a decision score for each of the matched classes as follows:

$$DS(C) = \sum_{i=1}^{k} Q(r_i),$$

[1] We use the C2 rule quality formula in the experiments described in the paper.

where r_i is a matched rule that indicates C, k is the number of these rules, and $Q(r_i)$ is the rule quality of r_i. The new example is then classified into the class with the highest decision score.

3. *No match.* The new example is not covered by any rule. Partial matching is considered where some attribute–value pairs of a rule match the values of corresponding attributes in the new example. If the partially–matched rules do not agree on classes, a partial matching score between an example e and a partially–matched rule r_i with n attribute–value pairs, m of which match the corresponding attributes of e, is computed as $PMS(r_i) = \frac{m}{n} \times Q(r_i)$. A decision score for a class C is computed as,

$$DS(C) = \sum_{i=0}^{k} PMS(r_i),$$

where k is the number of partially-matched rules indicating class C. In decision making, the new example is classified into the class with the highest decision score.

5 Classification Method

ELEM2 can learn rules from data with multiple classes and classify a new example into one of the classes. However, since a web page can belong to more than one category and the top frequent words for different categories are different, we transform our learning problem into multiple two-class learning problems. For each of 13 web page categories, we have a separate training data set, represented by top frequent words of the category. The web pages in the different training sets are the same among all the categories. For each category, we use ELEM2 to learn a binary classifier. When classifying a new page, 13 binary classifiers vote to determine which category or categories the new example belongs to. The voting method is as follows. We apply 13 classifiers to the new example to make binary decisions. The binary decisions are then combined by summing up the scores for each category. The category or categories that have the highest score are chosen to be the predicted category or categories for the new example. Table 2 shows the results from 13 binary classifiers and the voting result for a test page. In the table, C1, C2, ... and C13 denote categories, and B1, B2, ..., B13 denote binary classifiers. Since category C2 has most votes, the test page is classified into C2.

6 Evaluation Methods and Results

6.1 Evaluation Methods

Our objective is to investigate the effectiveness of our rough set feature selection method on web page classification. Two groups of tests are conducted in our experiments. In the first group of experiments, ELEM2 is used without using

Table 2. Classification and voting results for a test page.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
B1	0	1	1	1	1	1	1	1	1	1	1	1	1
B2	0	1	0	0	0	0	0	0	0	0	0	0	0
B3	1	1	0	1	1	1	1	1	1	1	1	1	1
B4	1	1	1	0	1	1	1	1	1	1	1	1	1
B5	1	1	1	1	0	1	1	1	1	1	1	1	1
B6	1	1	1	1	1	0	1	1	1	1	1	1	1
B7	1	1	1	1	1	1	0	1	1	1	1	1	1
B8	1	1	1	1	1	1	1	0	1	1	1	1	1
B9	1	1	1	1	1	1	1	1	0	1	1	1	1
B10	1	1	1	1	1	1	1	1	1	0	1	1	1
B11	1	1	1	1	1	1	1	1	1	1	0	1	1
B12	1	1	1	1	1	1	1	1	1	1	1	0	1
B13	1	1	1	1	1	1	1	1	1	1	1	1	0
Votes	11	13	11	11	11	11	11	11	11	11	11	11	11

rough set feature selection. In the second group, a reduct is computed for each of the training data sets. ELEM2 is then applied to the reduced sets of features to generate rules. In each group of experiments, to learn a binary classifier for each of 13 categories, we represent the training data using top n frequent words for that category. n is set to be 20, 30, 40, 50, and 60 in the experiments. Therefore, we have 13×5 training sets. Rules learned from each training set are then applied to examples in a test data set to make binary classifications for each example in the test set. Finally, the voting method is used to combine the binary classification results for each example into a final prediction for this example.

6.2 Test Data

The test data set contains a random sample of 370 web pages from each of 13 Yahoo categories (excluding the web pages used in the training phase). The total number of test pages is thus 4810. Each of these pages was turned into a separate testing example for each of 13 binary classifiers; the example is represented using the feature set corresponding to that classifier. Therefore, 13 binary classifications are made for a test web page. The final combined prediction for the web page is compared with the true membership of the page.

6.3 Performance Measures

In the machine learning community, it is common to consider the accuracy of a classifier on a test set as a good indication of classifier's performance. The testing accuracy is defined simply as the number of correct classifications divided by the total number of classifications. However, the text classification problem is different from typical machine learning problems in two aspects: examples may be given multiple class labels, which means separate binary classifiers must be trained for each class, and the positive examples of each class are usually in a very small minority [12]. These two characteristics imply that a plain accuracy statistic is not adequate to evaluate the performance of a text classifier because high accuracy can be achieved by always predicting the negative class in a binary classification. To deal with this unbalanced nature of classes, we use *precision* and *recall* instead of accuracy. In a binary classification system, *precision* is the proportion of examples labeled positive by the system that are truly positive, and *recall* is the proportion of truly positive examples that are labeled positive by the system.

In our web page classification, *precision* is the number of correct categories assigned divided by the total number of categories assigned, and serves as a measure of classification accuracy. The higher the precision, the smaller the amount of false categories. *Recall* is the number of correct categories assigned divided by the total number of known correct categories. Higher recall means a smaller amount of missed categories. To compute *precision* and *recall* for a test data set of web pages, we calculate the precision and recall for each example in the test set and then take the average values of precision and recall among all examples in the test set. Suppose the set of real categories of a test example is RC and the set of predicted categories is PC. Precision on this example is $\frac{RC \cap PC}{PC}$, and recall on this example is $\frac{RC \cap PC}{RC}$. Table 3 shows values of precision and recall for five sample test web pages.

Table 3. Precision and recall values for some test pages.

Page	Real Categories	Predicted Categories	Precision	Recall
P1	{2, 13}	{2}	1	1/2
P2	{8, 13}	{3, 8}	1/2	1/2
P3	{8, 9}	{2, 8, 9}	2/3	1
P4	{3}	{4, 6}	0	0
P5	{3, 4}	{3, 4}	1	1

6.4 Experimental Results

We applied the rough set feature selection method to each of 13×5 training data sets. Table 4 shows the number of eliminated attributes for each training set. The average number for each n, where n is the number of top frequent words used to represent the training data, is shown at the bottom of the table. The following observations are made. If top 20 words are used to represent the training data, no attributes are considered redundant by our rough set attribute reduction algorithm. Therefore, no attribute is removed. However, as more frequent words are used to represent the training data, more attributes are considered redundant and thus removed.

Table 4. Number of attributes eliminated by rough set feature selection.

Category	Top 20	Top 30	Top 40	Top 50	Top 60
1	0	0	2	2	2
2	0	0	0	0	0
3	0	0	0	1	0
4	0	1	1	2	4
5	0	1	1	4	5
6	0	1	2	1	4
7	0	0	0	0	4
8	0	1	1	1	1
9	0	0	0	1	3
10	0	0	0	0	1
11	0	0	2	1	5
12	0	2	2	2	3
13	0	0	0	1	0
Average	0	0.46	0.85	1.23	2.46

Figures 1 and 2 compare the classification results on the test data set in terms of precision and recall with and without rough set feature reduction. The results depend on the number of top frequent words used to represent training examples. If the number of original features is small (30 or 40), use of feature selection does not help in terms of precision. It actually decreases precision. On the other hand, when the number of original features becomes bigger (50 or 60), rough set feature selection improves the prediction precision. In terms of recall, in three of four categories of feature sets, an increase in recall is observed. Only for the top–40 category, we observe a decrease in recall by using the feature selection method. For the top–50 category, the increase is the most significant. Therefore, we can conclude that the rough set feature selection method is effective, that is, it can lead to better precision and recall, when the number of original features is large. One explanation for these results is that the larger the set of original features, the more likely it contains redundant or irrelevant features. Using rough set techniques, one can remove some redundant or irrelevant features and thus improve the predictive performance.

7 Conclusions

Automated categorization of web pages can lead to better web retrieval tools with the added convenience of selecting among properly organized directories. A challenge in automated web page classification or text classification in general is how to deal with the high dimensionality of the feature space. We have presented an application of machine learning techniques to web page classification. In particular, we used a feature selection method based on rough set theory to reduce the number of features used to represent a web page. We evaluated effectiveness

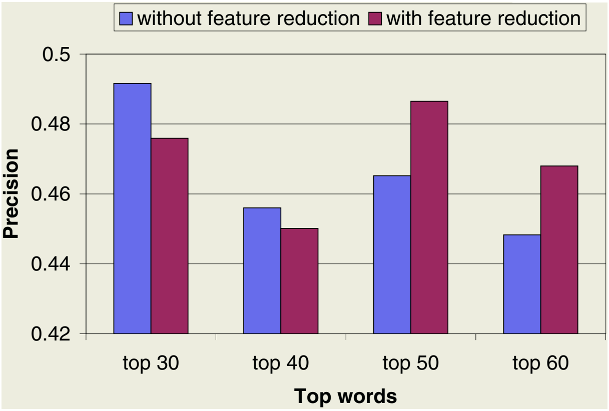


Fig. 1. Comparison in terms of precision.

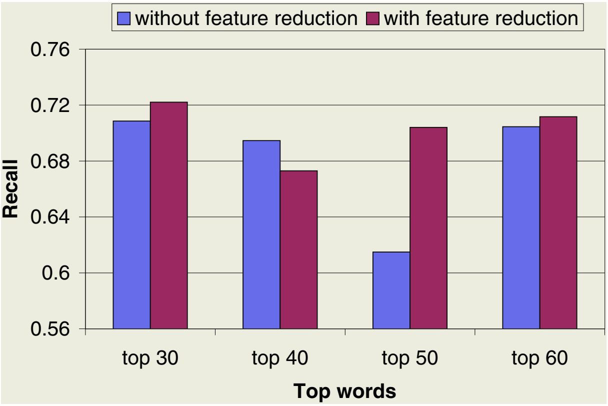


Fig. 2. Comparison in terms of recall.

of this rough set feature reduction method by comparing predictive performances of a learning system with and without this feature selection method. We observed that rough set feature selection method is effective when the set of features used to represent web pages is large. It can help to increase precision and recall by eliminating redundant or irrelevant features. When the feature set is small (under 50 top frequent words), the feature selection method may not help and may reduce the predictive performance. In our experiments, the rough set feature selection method is used as a pre–processing step for the ELEM2 learning system. However, it can be integrated with other learning methods to remove redundant and irrelevant features. In the future, we shall compare the rough set feature selection method with some statistical feature selection methods on web page classification.

Acknowledgments

This research was supported by grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Institute for Robotics and Intelligent Systems (IRIS).

References

1. An, A., Cercone, N. ELEM2: A Learning system for more accurate classifications, in: *Proceedings of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence* AI'98. Vancouver, Canada, 1998.
2. An, A., Cercone, N. Rule quality measures for rule induction systems: Description and evaluation. *Computational Intelligence*,**17:3**,2001, pp. 409–424.
3. Huang, Y. *Web-based Classification Using Machine Learning Approaches*. MSc Thesis, Department of Computer Science, University of Regina, Regina, Canada, 2002.
4. Lang, K. Newsweeder: Learning to filter netnews, in: *Proceedings of the Twelfth International Conference on Machine Learning.*
5. Lawrence, S., Giles, L. Accessibility and distribution of information on the Web. *Nature*,**400**, 1999, pp. 107–109. (http://www.metrics.com)
6. Lewis, D.D., Ringuette, M. Comparison of two learning algorithms for text categorization, in: *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval* SDAIR'94, 1994.
7. Mladenic, D. Feature subset selection in text learning, in: *Proceedings of the 10th European Conference on Machine Learning* ECML98, 1998.
8. Notess, G.R. Search engine statistics: database total size estimates. http://www.searchengineshowdown.com/stats/sizeest.shtml.
9. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer, 1991.
10. Porter, M.F. An algorithm for suffix stripping. *Program* **14:3**, 1980, pp. 130–137.
11. Raghavan, V.V., Sever, H. The state of rough sets for database mining applications, in: *Proceedings of 23rd Computer Science Conference Workshop on Rough Sets and Database Mining*, 1995, pp. 1–11.
12. Scott, S., Matwin, S. Text classification using WordNet Hypernyms, in: *Proceedings of the Conference on the Use of WordNet in Natural Language Processing Systems*,...1998,
13. Slowinski, R., ed. Intelligent Decision Support: Handbook of Advances and Applications of the Rough Sets Theory. Kluwer, 1992.
14. Wiener, E. Pedersen, J.O., and Weigend, A.S. A neural network approach to topic spotting, in: *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval* SDAIR'95, 1995.
15. Yang, Y. Noise reduction in a statistical approach to text categorization, in: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* SIGIR'95, 1995.
16. Yang, Y., Pedersen, J. A comparative study on feature selection in text categorization, in: *Proceedings of the 14th International Conference on Machine Learning* ICML'97, 1997, pp. 412–420.

On Learning Similarity Relations in Fuzzy Case-Based Reasoning

Eva Armengol, Francesc Esteva, Lluís Godo, and Vicenç Torra

Institut d'Investigació en Intel·ligència Artificial – CSIC
Campus UAB s/n, 08193 Bellaterra, Catalonia, Spain

Abstract. Case–based reasoning (CBR) is a problem solving technique that puts at work the general principle that similar problems have similar solutions. In particular, it has been proved effective for classification problems. Fuzzy set–based approaches to CBR rely on the existence of a fuzzy similarity functions on the problem description and problem solution domains. In this paper, we study the problem of learning a global similarity measure in the problem description domain as a weighted average of the attribute–based similarities and, therefore, the learning problem consists in finding the weighting vector that minimizes mis–classification. The approach is validated by comparing results with an application of case–based reasoning in a medical domain that uses a different model.

Keywords: case–base reasoning, fuzzy case–base reasoning, similarity relation, aggregation.

1 Introduction

Case–based reasoning, CBR for short, amounts to inferring that what is true in some known cases might still be true, possibly up to some suitable adaptation, in a newly encountered situation that is similar enough to those already known cases (see e.g. [1, 24]). In this way, case–based reasoning can be considered as a form of similarity–based or analogical reasoning since the basic principle implicitly assumed to apply in this kind of problem solving methodology is that *similar problems have similar solutions*.

Before going into more details, let us specify our working framework for classification–like case–based reasoning problems. Assume we have a base of cases CB consisting of an already solved set of cases, where a case is represented by a (complete) tuple of attribute values describing the situation or problem to solve together with a solution class or result. To fix ideas, let $\mathbf{A} = \{a_1, \ldots, a_n\}$ be the set of description attributes and let cl denote the class attribute. Moreover, let us denote by $D(a_i)$ and $D(cl)$ the domains of the attributes a_i and cl respectively (so $D(cl)$ is the set of solution classes). Then a case $c \in CB$ will be represented as a pair (d, r), where $d = (a_1(c), \ldots, a_n(c))$ is the set of the problem description values and $r = cl(c)$ is the solution class for the case c. If we write $\mathbf{D} = D(a_1) \times \ldots \times D(a_n)$, ($D$ for description) and $\mathbf{R} = D(cl)$, (R for result), then a case–base CB is just a subset of $\mathbf{D} \times \mathbf{R}$.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 14–32, 2004.

In this framework, given a case–base $CB = \{c_i = (d_i, r_i)\}_{i \in I}$, and a new problem description d_0, the CBR task is to find (guess) a solution class r_0 for d_0, by applying the above general principle in some form, i.e., by taking into account the possible similarity of d_0 to cases $c_i \in CB$ already solved.

It is clear then that the notion of *similarity* plays a key role in CBR problems. In particular, the notion of graded similarity, which has been used in the framework of fuzzy set theory for a long time [37], seems especially well suited for them. A fuzzy similarity relation on a domain Ω is a mapping $S : \Omega \times \Omega \to [0, 1]$ that assigns to every pair (w, w') of elements of Ω a number $S(w, w')$ measuring how much w and w' resemble each other according to some given criteria, in the sense that the higher $S(w, w')$, the more they resemble. In particular, $S(w, w') = 1$ means that w and w' are indistinguishable, while $S(w, w') = 0$ means that w and w' have nothing in common. One can also understand $\delta(w, w') = 1 - S(w, w')$ as a kind of distance between w and w'. Usual and reasonable properties required of such functions are reflexivity and symmetry, i.e., $S(w, w) = 1$ and $S(w, w') = S(w, w')$, for any $w, w' \in \Omega$. S is called *separating* if it verifies that $S(w, w') = 1$ iff $w = w'$. Sometimes, S is required additionally to fulfill a weak form of transitivity, viz., $S(w, w') \otimes S(w', w'') \leq S(w, w'')$, where $\otimes$ is a t-norm. For our purposes, and unless stated otherwise, we shall consider similarity relations as separating, reflexive and symmetric fuzzy binary relations.

In some recent literature a fuzzy set based approach to case–based reasoning has been developed not only from a practical point of view (see, e.g., [23, 7, 10, 6, 12, 21, 8]), but also from the formal modeling point of view [13, 35, 14, 30, 15, 16, 18, 9]. In all these models it is assumed that fuzzy similarity relations S_D and S_R on domains of problem description and solution attributes, $\mathbf{D}$ and $\mathbf{R}$, respectively, are known and given beforehand. In particular, several models have been proposed corresponding to different interpretations of the above CBR principle in terms of constraints between the fuzzy similarity relations S_D and S_R. In this paper, within the class of the so–called non–deterministic models, we tackle the problem of learning a particular type of global similarity measure from the set of precedent cases stored in the base case. Namely, given similarity measures S_a defined on each attribute domain $D(a)$, we show how to determine a weighting vector for each case in order to define a global similarity on $\mathbf{D}$ as a weighted average of the S_a's that minimizes the mis–classifications. We check their adequacy by comparing results with an application of case–based reasoning in a medical domain that uses a different model. In that model cases are not attribute–value tuples but they are represented in a relational way. A relational representation describes objects on the basis of their components and relations between these components. In Machine Learning there is a wide research field, Inductive Logic Programming, focused on relational representation of objects and methods to handle them (see e.g. [28]).

The paper is organized as follows. In Sect. 2 we describe several fuzzy CBR models. Then, in Sect. 3 we show how to construct fuzzy similarity relations to be used in the fuzzy CBR models. Sect. 4 explains an alternative approach using symbolic similarity for comparing relational cases. The results of both approaches are compared in Sect. 5. The paper finishes with the conclusions.

2 Fuzzy CBR Models

A first class of models try to model a strong form of the case–based reasoning principle which reads:

> "the more similar are the description values (in the sense of S_D),
> the more similar are the class values (in the sense of S_R)".

In [13, 14], this principle is modeled by regarding each case $c = (d, r)$ as a *fuzzy gradual rule* of the form

If X is *approximately_d* then Y is *approximately_r*.

Here X is a linguistic variable on the domain $\mathbf{D}$ of problem description tuples and Y is a linguistic variable on the domain $\mathbf{R}$ of solution classes, and *approximately_d* and *approximately_r* denote fuzzy sets of attribute value tuples close or similar to d and r, and defined on $\mathbf{D}$ and $\mathbf{R}$, respectively, by

$$\mu_{approximately\_d}(d') = S_D(d, d'), \quad \mu_{approximately\_r}(r') = S_R(r, r').$$

The semantics of fuzzy gradual rules (see [17]) capture the above intended meaning in the sense that the conditional possibility distribution they induce on $\mathbf{D}\times\mathbf{R}$ is,

$$\pi(r' \mid d') \leq \left[\mu_{approximately\_d}(d') \Rightarrow_{\otimes} \mu_{approximately\_r}(r')\right],$$

where $\Rightarrow_{\otimes}$ is the residuum of a continuous t-norm $\otimes$. Such a binary operation on [0, 1] has as fundamental property (independently of $\otimes$) that $\left[x \Rightarrow_{\otimes} y\right] = 1$ iff $x \leq y$. This means that, given a current problem d_0, the best solutions for d_0 which can be inferred from the above rule are those r_0 such that,

$$\left[\mu_{approximately\_d}(d_0) \Rightarrow_{\otimes} \mu_{approximately\_r}(r_0)\right] = 1,$$

or equivalently such that,

$$S_D(d, d_0) \leq S_R(r, r_0).$$

That is, the best solutions are those r_0 with a similarity to r that is at least as much as the similarity of d_0 to d. So, with this interpretation, neighborhoods around d are transferred to neighborhoods around r. Now, if we consider the whole case base $CB = \{c_i = (d_i, r_i)\}_{i\in I}$, and we perform the same kind of inference for each case (d_i, r_i), then the global solution set will be the conjunctive aggregation of the individual sets,

$$\bigcap_{i\in I}\{r_0 \in \mathbf{R} \mid S_D(d_i, d_0) \leq S_R(r_i, r_0)\}.$$

One problem here is that, in principle, nothing prevents that this intersection can be empty, if the case base CB is not fully consistent with the above principle. In particular this can happen when the case base CB contains cases with very similar description attribute values but with different class values. This is why such a model was called *deterministic* in [13, 14].

In [18] alternative interpretations of the same principle were considered, namely by assuming the inequality,

$$\text{if} \quad S_D(d_1, d_2) \leq S_D(d_1, d_3) \quad \text{then} \quad S_R(r_1, r_2) \leq S_R(r_1, r_3),$$

(and two other similar ones) to hold for any triple of cases $c_1 = (d_1, r_1), c_2 = (d_2, r_2), c_3 = (d_3, r_3) \in CB$. But still, these are deterministic models in the above sense, i.e $S_D(d_1, d_2) = 1$ implies $S_R(r_1, r_2) = 1$.

Deterministic models can be felt too strong in many real-world domains, since in some sense they assume that the description attributes completely characterize all possible situations. This is not often the case. In contrast the so-called *non-deterministic* models assume a weaker form of general CBR principle, in the sense that "it is only plausible (not necessary) that similar problems have similar solutions".

Expressed in terms of the fuzzy similarity relations S_D and S_R, this weaker Case-Based Reasoning principle can be expressed by the following rule [13, 14]:

> "The more similar are the problem descriptions in the sense of S_D, the more *possible* the solution classes are similar in the sense of S_R."

This amounts to state that for any case (d, r) in the case base CB, the closer is the current problem d_0 to d, the more plausible r is a solution class for d_0. This can be formalized again as a fuzzy rule,

$$\text{If } X \text{ is } \mathit{approximately\_d} \text{ then } Y \text{ is } \mathit{approximately\_r},$$

but with a different semantics, corresponding to the so-called *possibility fuzzy rules* (see [17]). The semantics of such a possibility fuzzy rule is "the more X is *approximately_d* (i.e. the closer is a description to d), the more possible (plausible) is *approximately_r* a range for Y (i.e. the more plausible are those solutions close to r)". In terms of possibility distributions, the above rule induces the following constraint on the conditional possibility distribution on $\mathbf{D} \times \mathbf{R}$:

$$\pi(y \mid x) \geq \min(\mu_{approximately\_d}(x), \mu_{approximately\_r}(y)).$$

In other words, the plausibility degree of y being a solution for the problem x is bounded from below by the similarity degree between y and r, truncated by the degree to which x is similar to s. When we consider the whole case–base $CB = \{c_i = (d_i, r_i)\}_{i \in I}$, then the joint constraint induced by all fuzzy rules corresponding to the cases in CB is the disjunctive aggregation of the individual ones:

$$\pi(y \mid x) \geq \max_{(d_i, r_i) \in CB} \min(\mu_{approximately\_d_i}(x), \mu_{approximately\_r_i}(y)).$$

Then, given a current description problem d_0, such a joint constraint induces a *plausibility ordering* π_{d_0} on possible solutions or classes for d_0 by,

$$\pi_{d_0}(y) = \max_{(d_i, r_i) \in CB} \min(\mu_{approximately\_d_i}(d_0), \mu_{approximately\_r_i}(y)) =$$

$$\max_{(d_i,r_i)\in CB} \min(S_D(d_i,d_0), S_R(r_i,y)),$$

for all $y \in \mathbf{R}$, in the sense that the greater $\pi_{d_0}(y)$, the more plausible y as a solution for d_0. Sometimes, plausibility values in themselves are not particularly important but only the ordering they induce on the set $\mathbf{R}$ of solution classes:

$$y \preceq_{d_0} y' \text{ iff } \pi_{d_0}(y) \leq \pi_{d_0}(y').$$

Finally, notice that if new cases are added to the case base CB, $\pi_{d_0}(y)$ can only increase, and never decrease, according to the idea that new cases may incorporate new solutions but not discard old ones. Non–deterministic models have been also considered for instance in [22, 10].

3 Learning Similarity Relations for a Non-deterministic CBR Model

As already mentioned in Introduction, in this chapter we are concerned with learning aspects of similarities within a non–deterministic fuzzy set approach to CBR, with some particular choices. In this section, we first set our working assumptions for the model, and then we describe how the relevant similarity relations can be constructed from the case–base.

3.1 Working Assumptions and Relevant Similarity Relations

Background knowledge is assumed for each attribute $a \in \mathbf{A}$ under the form of a similarity relation S_a on $D(a)$, the domain of a. This similarity is a function $S_a : D(a) \times D(a) \to [0,1]$. Additionally, we assume a similarity relation S_{cl} defined over the set of classes. This similarity was denoted by S_R in the previous section but from now on we use for it the symbol S_{cl} to stress the fact that it is defined over the set of classes. This corresponds to a function $S_{cl} : D(Cl) \times D(Cl) \to [0,1]$.

The goal is to define a similarity relation S_D among the cases in the case–base CB and an arbitrary problem description. Our working assumption is that such similarity will be defined as a weighted average of the existing similarity functions S_a for each attribute. Of course, we then need additional information to assess the relevance of each attribute for retrieving a particular case. In particular, we shall assume there is a *weighting vector for each case* that evaluates the importance of each attribute when computing the similarity between this case and another arbitrary case description in the case–base CB. This is formalized in the following definition.

Definition 1. *Let $CB \subset \mathbf{D} \times \mathbf{R}$ be a case–base, $\mathbf{A}$ the set of attributes considered in $\mathbf{D}$, and, for each $a \in \mathbf{A}$, let S_a be the corresponding similarity relation on $D(a)$. Finally, let w_c be the weighting vector*[1] *attached to a particular case*

[1] $w = (w_1, \cdots, w_n)$ is a weighting vector of dimension n iff $w_i \geq 0$ for all $i = 1, ..., n$, and $\sum_{i=1,n} w_i = 1$.

$c \in CB$. Then, the similarity relation value defined for an arbitrary case description $d \in \mathbf{D}$ and c is,

$$S_D(c, d) = \sum_{a \in \mathbf{A}} w_c(a) \cdot S_a(a(c), a(d)). \tag{1}$$

As $S_a(\cdot, \cdot) \in [0, 1]$ and as w_{cb} is a weighting vector, it follows that $S(\cdot, \cdot)$ is a function into the interval $[0, 1]$ as well.

Note that, formally, S_D so defined is a function on $CB \times \mathbf{D}$, although for cases in CB we only make use of their description attribute values. Also, this definition is not commutative in the sense that it is the case c who determines the weighting vector (the weighting vectors may be different for two different cases), so the first argument in S_D plays a preeminent role.

Once we have defined the similarity relation S_D, we can define how close is a problem description to a solution class just by comparing it to every case in CB that shares that solution class.

Definition 2. *Let CB, $\mathbf{D}$, $\mathbf{A}$ and S_D be defined as in Definition 1. Then, the similarity between a case description $d \in \mathbf{D}$ and a solution class $r \in D(cl)$ is defined as follows:*

$$SCL(d, r) = \max_{c \in \mathbf{CB}, cl(c) = r} S_D(c, d).$$

In turn, by this definition, we can define similarity between arbitrary pair of case descriptions as a kind of transitive closure of the SCL function as follows.

Definition 3. *Let CB, $\mathbf{D}$, $\mathbf{A}$, S_D and SCL be defined as in Definition 2. Let $\otimes$ be a continuous t-norm. Then, the $\otimes$-similarity between two arbitrary set of case descriptions $d_1, d_2 \in \mathbf{D}$ is defined as follows:*

$$SCC(d_1, d_2) = \sum_{r \in D(cl)} SCL(d_1, r) \otimes SCL(d_2, r).$$

Among others, possible choices for $\otimes$ are e.g. $\otimes = \cdot$ or $\otimes = \min$.

3.2 Construction of the Numerical Similarity Function S_D

Definition 1 introduces the similarity value $S_D(c, d)$ between an arbitrary problem description d and a case c in the base case CB. Recall that this similarity has been defined as the weighted mean of the attribute similarities S_a for a in $\mathbf{A}$. However, expression (1) makes it explicit that the function is not symmetric in the sense that the two arguments are not equally important. Instead, the role of the case c is preeminent because we make the weighting vector w_c depending on c. Next, we describe how to learn the appropriate weights for each $c \in CB$.

In the following we assume that a case $cb \in CB$ is known and fixed along the learning process. In fact, the same process we describe below for cb will be applied for each case in CB. Naturally, for each case cb, the process would lead to its corresponding weighting vector w_{cb}.

To compute the vector, besides of the case *cb*, we need to fix a subset of the case–base, i.e., a collection of problem descriptions for which their solution class is known. This is the learning set, denoted by LS. Of course, we shall also make use of the similarities S_a for each attribute $a \in \mathbf{A}$.

Then, weights determination can be formulated in the following way.

Problem 1 (Weight Determination Problem). Let LS be the learning set and $cb \in CB$ a case in the case–base. Weight Determination Problem is to find the set of weights w_{cb} in (1) such that, for each $c = (d, r) \in LS$, the similarity value between cb and d, $S_D(cb, d)$, approximates as much as possible the similarity between the solution classes $cl(cb)$ and $cl(c) = r$, $S_{cl}(cl(cb), r)$.

Using the square difference to measure the divergence between the two similarities (i.e., the similarity between the two cases and the similarity between their classes), we can re–formulate the problem as follows:

Problem 2. Let $LS = \{c_j\}_{j \in J}$ be the learning set and cb a case in the case–base. Then Weight Determination Problem relative to cb is to find the set of weights $w_{cb}(a_1), ..., w_{cb}(a_n)$ that minimizes the expression,

$$\sum_{c_j \in LS} \big(\sum_{a_i \in \mathbf{A}} w_{cb}(a_i) \cdot S_{a_i}(a_i(c_j), a_i(cb)) - S_{cl}(cl(c_j), cl(cb)) \big)^2,$$

subject to the following constraints over w_{cb},
(1) $\sum_{a_i \in \mathbf{A}} w_{cb}(a_i) = 1$, and
(2) $w_{cb}(a_i) \geq 0$ for all $a_i \in \mathbf{A}$.

Now, we introduce simplified terms for similarities so that the problem can be further simplified. The notation we will use is,

- $a_i^j = S_{a_i}(a_i(c_j), a_i(cb))$ is the similarity between the values corresponding to i-th attribute in $\mathbf{A}$ for the case cb and the j-th case in LS;
- $b^j = S_{cl}(cl(c_j), cl(cb))$ is the similarity between the solution classes of cb and the j-th case in LS;
- $w_i = w_{cb}(a_i)$ corresponds to the weight of the attribute a_i.

In this way, the data available to the problem has the form of Table 1 and Weights Determination Problem corresponds to,

Problem 3.

$$\begin{array}{l} Minimize\ \sum_{j=1}^{|LS|} \left(\sum_{i=1}^{|\mathbf{A}|} w_i a_i^j - b^j \right)^2 \\ Subject\ to: \\ \qquad - \ \sum_{i=1}^{|A|} w_i = 1 \\ \qquad - \ w_i \geq 0, \text{ for all } i = 1, ..., |A| \end{array}$$

The minimization problem formulated above has been studied in [19], [32] and [33] in the general framework of parameter learning for aggregation operators. [19] introduced a method based on the gradient descent for learning the

Table 1. Structure of the available data for finding the weighting vector.

S_{a_1}	S_{a_2}	$\cdots$	$S_{a_{\lvert\mathbf{A}\rvert}}$	S_{Cl}
a_1^1	a_2^1	$\cdots$	$a_{\lvert\mathbf{A}\rvert}^1$	b^1
a_1^2	a_2^2	$\cdots$	$a_{\lvert\mathbf{A}\rvert}^2$	b^2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$a_1^{\lvert LS\rvert}$	$a_2^{\lvert LS\rvert}$	$\cdots$	$a_{\lvert\mathbf{A}\rvert}^{\lvert LS\rvert}$	$b^{\lvert LS\rvert}$

weights of the OWA operator [36]. The similarities between the OWA and the Weighted Mean make the method equally suitable for the Weighted Mean. In [32] an algorithm based on Active Set Methods was studied and applied to some medium–size data sets (problems up to 34 attributes were considered). Results reported were positively compared with the ones in [19]. More recently, [33] studied some issues left open in [32]. In particular, the work studied the situation with linearly dependent attributes. Then, an algorithm was introduced to deal with such situation. In this work we apply the algorithms introduced in [32] with the extension described in [33]. Related results about parameter learning for aggregation operators include [20], [26] and [31]. By the way, these latter results have been defined for learning parameters of more complex operators (e.g., the Choquet integral).

Methods based on active sets (see e.g. [29]) rely on the simplicity of computing the solution of quadratic problems with linear equality constraints. Iterative algorithms have been developed in which at each step inequality constraints are partitioned into two groups: those that are to be treated as active (considered as equality constraints) and those inactive (essentially ignored). Once a partition is known, the algorithm proceeds, by moving on the surface defined by the working set of constraints (the set of active constraints), to an improved point. In this movement, some constraints are added to the working set and some others are removed. This process is repeated until the minimum is reached. When the function to be minimized is convex (this is the case for the weighted mean), the method finds the minimum and, although the method is iterative, the final minimum is not influenced by the initial weight vector.

However, an important problem arises in practical applications of these methods: some data can have linearly dependent columns (in our case, this corresponds to non–independent attributes). In this case, the algorithm fails to give a solution. In fact the problem arises only, as shown in [33], when there is a column/attribute a_i that can be written as a linear combination of the others of the form $a_i = \sum_j p_j a_j$ in such a way that $\sum_j p_j = 1$. In such case, when removing one of the linearly dependent columns, we get the same minimum we would get when considering all the attributes. Therefore, an alternative approach is to consider as many sub–problems as dependent attributes, where each sub–problem corresponds to the original one after removing one of the dependent attributes. The solution with a minimum error would correspond to the solution of the original problem.

4 A Symbolic Similarity Approach to Relational CBR

Reasoning and learning from cases is based on the concept of similarity, this is clear, but there exist several ways of modeling similarity. Numerical (or fuzzy) similarity–based approaches to case retrieval, like the ones we have considered in previous sections, are mainly used for cases represented as attribute–value vectors. Instead, in this section we consider another approach where cases are represented in a scheme that uses relations among entities. We will call this setting *relational case–based learning.* In this approach, the similarity between two cases is understood as what they "share". But, in addition, we need to be able to evaluate whether what they share is "relevant" or "important" (or to what degree it is relevant or important) for the problem at hand. In this section, we first introduce concepts of symbolic similarity and feature terms, the basis of the formalism we will use for representing relational cases. Then, we present LID (for Lazy Induction of Descriptions), a method for relational case–based learning. LID, introduced in [5], is based on two main notions: 1) the similarity is constructed as a symbolic description of what is shared between precedent cases and a specific *problem* to be classified, and 2) there is some assessment function to help the system decide which relations among entities are important or relevant to be shared with the precedent cases.

4.1 Symbolic Similarity and Feature Terms

In real domains it can be useful to represent knowledge in a structured way. As a means to do so, in [4] the authors introduced the so–called *feature terms* (also called feature structures or ψ-terms) that are a generalization of first order terms [2, 11]. Formally, we define a feature term as follows.

Definition 4. *Given a signature $\Sigma = \langle \mathcal{S}, A, \leq \rangle$, where $\mathcal{S}$ is a set of sort symbols, $\mathbf{A}$ is a set of attribute symbols and $\leq$ is a decidable partial order on $\mathcal{S}$, and a set ϑ of variables, we define a* feature term *as an expression of the form,*

$$\psi ::= X : s[a_1 \doteq \Psi_1 \dots a_n \doteq \Psi_n], \tag{2}$$

where X is a variable in ϑ called the root *of the feature term, s is a sort in $\mathcal{S}$, $a_1 \dots a_n$ are attributes in $\mathbf{A}$, and each Ψ_i is a set of feature terms and variables. When $n = 0$, we are defining a variable without features. The set of variables occurring in ψ is noted as ϑ_ψ.*

Figure 1 shows the description of a diabetic patient using feature terms. The patient *patient-371* is a feature term of sort *patient* (sorts are underlined in the figure). This patient is described by two attributes, *diabetes-data* and *consultation*. The value of *diabetes-data* is a feature term of sort *diab-data* that has, in turn, two attributes, *dm-type* and *dm-year*.

Definition 5. *A path $path(X, a_i)$ is defined as a sequence of attributes going from the variable X to the feature a_i.*

$$
\textit{patient-371} = \left[\begin{array}{l}
\underline{\textit{patient}} \\
\textsf{diabetes-data} \doteq \left[\begin{array}{l} \underline{\textit{diab-data}} \\ \textsf{dm-type} \doteq 2 \\ \textsf{dm-year} \doteq 1998 \end{array}\right] \\
\textsf{consultation} \doteq \left[\begin{array}{l}
\underline{\textit{consultation}} \\
\textsf{additional-treatment} \doteq \left[\begin{array}{l} \underline{\textit{additional-treatment}} \\ \textsf{hypertension} \doteq false \\ \textsf{dyslipidaemia} \doteq true \\ \textsf{other-t} \doteq false \end{array}\right] \\
\textsf{eye-exam} \doteq \left[\begin{array}{l} \underline{\textit{eye}} \\ \textsf{visibility} \doteq \left[\begin{array}{l} \underline{\textit{visibility}} \\ \textsf{maculopathy} \doteq false \\ \textsf{retinopathy} \doteq false \end{array}\right] \\ \textsf{photocoagulation} \doteq false \\ \textsf{cataract} \doteq false \end{array}\right] \\
\textsf{measurements} \doteq \left[\begin{array}{l} \underline{\textit{measurements}} \\ \textsf{macro-compl?} \doteq false \\ \textsf{q-hba1c} \doteq low \\ \textsf{blindness} \doteq false \\ \textsf{q-bp} \doteq low \\ \textsf{smoker?} \doteq false \\ \textsf{q-albumin} \doteq low \\ \textsf{q-cholesterol-total} \doteq high \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Fig. 1. Partial description of a diabetic patient. Here only 22 attributes are shown out of a total of more than 80 paths present in the complete description of the patient.

Some examples of paths in the description of figure 1 are the following:

- $path(\textit{patient-371}, retinopathy)$
  $= \textit{patient-371}.$consultation.eye-exam.visibility.retinopathy;
- $path(\textit{patient-371}, dyslipidaemia)$
  $= \textit{patient-371}.$consultation.additional-treatment.dyslipidaemia;
- $path(consultation, visibility) =$ consultation.eye-exam.visibility.

Sorts have an informational order relation ($\leq$) among them, where $s \leq s'$ means that s is more general than s' (or s' has more information than s), we also say then that s' is a sub–sort of s. When a feature has *unknown* value it is represented as having the sort *any*. All other sorts are more specific than *any*.

There is an ordering relation, called *subsumption*, among feature terms defined as follows:

Definition 6. *A feature term ψ* subsumes *another feature term ψ', written $\psi \sqsubseteq \psi'$, when the following conditions are satisfied: 1) the sort of ψ' is either the same or a sub–sort of the sort of ψ, 2) if A_ψ is the set of attributes of ψ and $A_{\psi'}$ is the set of attributes of ψ' then $A_\psi \subseteq A_{\psi'}$ and 3) the feature terms of values of the attributes in A_ψ and $A_{\psi'}$ satisfy in turn the two conditions above.*

```
Function $LID(d, S_t, \Delta_S, R)$
    if stopping-condition($\Delta_S$)
      then return $Class(\Delta_S)$
      else $a_d$ := Select-attribute ($d$, $\Delta_S$, $R$)
          $S'_t$ := Add-path($path(d, a_d)$, $S_t$)
          $\Delta_{S'}$ := Discriminatory-set ($S'_t$, $\Delta_S$)
          $LID(d, S'_t, \Delta_{S'}, R)$
    end-if
end-function
```

Fig. 2. The *LID* algorithm. S_t is the similarity term, Δ_S is the discriminatory set associated with S_t, R is the set of solution classes, $class(\Delta_S)$ is the class $r_i \in R$ to which belong all the elements in Δ_S.

Intuitively, a feature term ψ subsumes ψ' when all information in ψ is also contained in ψ'.

Definition 7. *A feature term S_t is a* similarity term *of two cases c_1 and c_2 if and only if $S_t \sqsubseteq c_1$ and $S_t \sqsubseteq c_2$, i.e., the similarity term of two cases subsumes both cases. The set Δ_S of cases subsumed by the similarity term S_t is called the* discriminatory set *associated with S_t.*

As we will see in Section 4.2, a similarity term can be seen as a symbolic similarity among cases. In short, given a problem description d, *LID* classifies it as belonging to the solution class $r \in \mathbf{R}$ when all the cases in the discriminatory set Δ_S associated with the similarity term S_t have r as solution class. This assumption can be made because *LID* builds the similarity term taking into account the attributes that are more relevant for classifying the problem d. Intuitively, *LID* considers that if the problem description d shares a set of relevant attributes (those in the similarity term S_t) with a (sub)set of cases belonging all of them to the same solution class r, then d can also be classified as belonging to r. In Section 4.2 we explain how to select the relevant attributes of a case to form the similarity term.

4.2 The LID Method

In this section we describe the method LID, useful for relational case–based learning. LID combines the notion of symbolic similarity with an heuristics that measures the importance of an attribute as the distance between the partition induced by the values of this attribute and the correct one. The criterion used in LID is based on the discrimination power of the similarity term which is evaluated using the so-called RLM distance [25].

The main steps of the *LID* algorithm are shown in Figure 2. Input parameters of the *LID* algorithm are a problem description d to be classified, a similarity term S_t, the discriminatory set Δ_S associated with S_t and the set of solution classes $\mathbf{R} = D(cl)$ where the problem d can be classified. In the first call $LID(d, S_t, \Delta_S, \mathbf{R})$, the similarity term S_t is initialized to the most general description (the one having no features) and the set $\varDelta_S$ is initialized to CB, the whole case base.

The stopping condition of LID is when all the cases in the discriminatory set $\varDelta_S$ belong to only one solution class $r_0 \in R$. In such a situation LID gives the similarity term S_t as an explanation of the classification of d in r_0 and $\varDelta_S$ is the support set justifying that result[2]. The similarity term S_t can be viewed as a *partial* description of the solution class r_0 because it contains a subset of attributes that are discriminating enough to classify a case as belonging to r_0. Notice however that S_t is not the most general "generalization" of r_0, since in general S_t does not subsume all the cases belonging to r_0 but only a subset of them (those sharing the attributes of S_t with the new problem). The similarity term S_t depends on the current problem, for this reason there are several partial descriptions (i.e. similarity terms) for the same solution class.

In the first call of LID, the stopping condition is not satisfied since $\varDelta_S$ contains all the cases in CB. This means that the similarity term S_t is satisfied by cases belonging to several solution classes, therefore S_t has to be specialized.

The specialization of a similarity term S_t is achieved by adding attributes to it. LID considers as candidates to specialize S_t only those attributes that are present in the problem description d to be classified. Let A_l be the set of attributes in d with no *unknown* value.

The next step of LID is the selection of an attribute $a_d \in A_l$ to specialize the similarity term S_t. The selection of the most discriminatory feature in the set A_l is done heuristically by using the RLM distance over the features in A_l.

The RLM distance assesses how similar are two partitions (in the sense that the smaller the distance, the more similar they are). Each attribute $a_i \in A$ induces a partition on the case–base, where classes sp_{ij} are formed by cases having the same value for the attribute a_i. The *correct partition* is the partition $\mathcal{P}_c = \{\bar{r}_1, \ldots, \bar{r}_m\}$, where the classes correspond to the solution classes, i.e. the cases belonging to $\bar{r}_i$ are those with the same solution class r_i. For each partition $\mathcal{P}_i$ induced by an attribute a_i, LID computes its RLM distance to the correct partition $\mathcal{P}_c$. The proximity of the partition $\mathcal{P}_i$ to $\mathcal{P}_c$ estimates the relevance of feature a_i.

Definition 8. *Given two partitions $\mathcal{P}_i = \{sp_{i1}, \ldots, sp_{in}\}$ and $\mathcal{P}_c = \{\bar{r}_1 \ldots \bar{r}_m\}$ of the case base CB, the RLM distance between them is computed as,*

$$RLM(\mathcal{P}_i, \mathcal{P}_c) = 2 - \frac{I(\mathcal{P}_i) + I(\mathcal{P}_c)}{I(\mathcal{P}_i \cap \mathcal{P}_c)},$$

with

$$I(\mathcal{P}_i) = -\sum_{j=1}^{n_i} p_j \cdot log_2 p_j \quad ; \quad p_j = \frac{|CB \cap sp_{ij}|}{|CB|};$$

[2] There is an *abnormal* stopping condition when there cases in $\varDelta_S$ belonging to different solution classes but the similarity term S_t cannot be further specialized. In such a case, the output of LID is not a single class but the set of solution classes of the cases in $\varDelta_S$.

$$I(\mathcal{P}_c) = -\sum_{k=1}^{m} p_k \cdot log_2 p_k \quad ; \quad p_k = \frac{|CB \cap \bar{r}_k|}{|CB|};$$

$$I(\mathcal{P}_i \cap \mathcal{P}_c) = -\sum_{j=1}^{n_i}\sum_{k=1}^{m} p_{jk} \cdot log_2 p_{jk} \quad ; \quad p_{jk} = \frac{|CB \cap \bar{r}_k \cap sp_{ij}|}{|CB|},$$

where $I(\mathcal{P}_i)$ measures the information contained in the partition $\mathcal{P}_i$; n_i is the number of possible values of the attribute inducing $\mathcal{P}_i$; p_j (p_k resp.) is the probability of occurrence of the class sp_{ij} ($\bar{r}_k$ resp.), i.e., the proportion of examples in CB that belong to sp_{ij} ($\bar{r}_k$ resp.); $m = |\mathcal{P}_c|$, i.e., the number of solution classes; $I(\mathcal{P}_i \cap \mathcal{P}_c)$ is the mutual information of the two partitions; and p_{jk} is the probability of the intersection $\bar{r}_j \cap sp_{ik}$, i.e., the proportion of examples in CB that belong to $\bar{r}_j$ and to sp_{ik}. In this definition, as it is common in the case of entropy, $0 \log 0$ *is defined as zero.*

Definition 9. *Let $\mathcal{P}_i$ and $\mathcal{P}_j$ be partitions induced by attributes a_i and a_j respectively. We say that a_i is* more discriminatory than *a_j iff $RLM(\mathcal{P}_i, \mathcal{P}_c) < RLM(\mathcal{P}_j, \mathcal{P}_c)$, i.e., when the partition induced by a_i is closer to the correct partition $\mathcal{P}_c$ than the partition induced by a_j.*

LID uses the *more discriminatory than* relationship to estimate the attributes that are more relevant for the purpose of classifying a current problem. Let us call a_d the most discriminatory attribute in $\mathbf{A}$, i.e., a_d induces the partition of the discriminatory set closest to the correct partition.

The specialization step of *LID* defines a new similarity term S'_t by adding to the current similarity term S_t the sequence of attributes specified by $path(d, a_d)$. After this addition S'_t has a new path with all the attributes in the path taking the same value they take in d. After adding the path $path(d, a_d)$ to S_t, the new similarity term $S'_t = S_t + path(d, a_d)$ subsumes a subset of cases in Δ_S, namely the discriminatory set $\Delta_{S'}$.

Next, *LID* is recursively called with the similarity term S'_t and the discriminatory set $\Delta_{S'}$. The recursive call of *LID* has $\Delta_{S'}$ as parameter (instead of Δ_S) because the cases that are not subsumed by S'_t will not be subsumed by any further specialization. The process of specialization reduces the discriminatory set $\Delta_{S^n} \subseteq \Delta_{S^{n-1}} \subseteq \ldots \subseteq \Delta_{S'} \subseteq \Delta_S$ at each step. This specialization of the discriminatory set will provoke the algorithm to terminate.

5 Application

In this section, we describe the results obtained when applying the non–deterministic fuzzy CBR approach, with the fuzzy similarity learning method described in Section 3, into a medical domain (in a Diabetis related problem). Similarity relations have been built for a set of cases and results are reported and compared with those obtained by the LID method. Additionally, we describe the classification results obtained when using the *leave one out* approach. The section starts with the description of the problem and follows with the description and analysis of the experiments.

5.1 The Diabetes Domain

Diabetes Mellitus is one of the most common human chronic diseases. There are two major types of diabetes: diabetes type I (or *insulin–dependent*) and diabetes type II (or *non–insulin-dependent*). The diabetes type I is usually found in people under 40 years and is the consequence of a pancreatic malfunction. Instead, diabetes type II is more frequent in aged people and it could be explained as a loss of effectivity of the insulin on the body cells. In fact, both forms of diabetes produce the same short–term symptoms (i.e., frequent urination, increase of thirst and high blood glucose values) and long–term complications (i.e., blindness, renal failure, gangrene, coronary heart disease and stroke). The main concern in the management of diabetes is to reduce the risk of the patient to develop long–term complications. In 1989, representatives of Government Health Departments, patient organizations and diabetes experts celebrated a meeting in Saint Vincent (Italy) (http://www.show.scot.nhs.uk/crag/topics/diabetes/vincent.htm) to elaborate some recommendations to be followed in the diabetes management with the goal of minimizing the diabetic complications of the patients.

Since then, experts have analyzed a lot of data that allowed to define the main parameters (attributes) that prevent the development of complications. In particular, they found that keeping the analytical data of the patient as close as possible to the "normal" ranges, clearly reduces the complication risk. In other words, the patient has to modify some of his life habits (like diet, physical exercise, alcohol ingestion or smoking habits) in order to maintain analytical parameters, such as the cholesterol, the blood pressure or the creatinine, within the same ranges than those of a healthy person. Furthermore, it is necessary a strict eye and foot control to prevent the development of a retinopathy and of a polyneuropathy respectively.

In this work we have considered the assessment of infarct risk of a diabetic patient. Four solution classes are considered: *low, moderate, high* and *very-high.* The assessment of the risk can be seen as a classification task where the goal is to identify the solution class to which the new problem belongs. In the next section we describe the process of building a similarity function to determine the attributes that are more important in order to assess the infarct risk of a diabetic patient. An evaluation of the built similarity function is also considered.

5.2 Empirical Description and Analysis

As briefly described above, the procedure explained in Section 3.2 has been applied to find similarities between cases. We have considered a case base CB with 30 cases describing patient analytical data and patient's risk assessment (real analytical data was supplied by the Mataró Hospital and the risk of each patient was assessed by a physician). Each case was described in terms of 84 attributes.

Then, the procedure outlined in Section 3.2 for the determination of the weighting vectors has been applied. In this way, we have learned a weighting vector for each case $cb \in CB$. To do so, we have used as learning set the whole

CB after removing the case cb. This is, for each $cb \in CB$, $LS_{cb} = CB - \{cb\}$. The learning stage lead to 30 weighting vectors, each one of dimension 84. This process required first the definition of similarity functions S_{cl} and S_a for all $a \in \mathbf{A}$.

Following the process explained in Section 3.2 we built a matrix similar, in structure, to the one in Table 1. It has to be mentioned that at that point, and before applying the learning algorithm, we considered to run the experiment with both the original data and normalized data. In the second case, each variable was normalized in [0,1] defining

$$a'^i_j = \frac{(a^i_j - min_k a^i_k)}{(max_k a^i_k - min_k a^i_k)} .$$

The normalization was considered because the similarity range of some variables never reached 0 or 1 for the cases considered in the learning set.

Linear dependency problems raised in the computation of the weights for 15 of the cases in CB (for both normalized and non-normalized data). In these cases, we proceeded, as explained in Section 3.2, to remove one dependent attribute at a time and computing the weights for all these subproblems. For all of these subproblems, the best weighting vector was selected. Nevertheless, some of the problems still had linear dependent attributes. For instance, the 6–th and 7–th cases had more than two dependent columns when normalized data was considered, and the same occurred for the 7–th case with the non–normalized data. No additional treatment was applied to these cases and thus the best weighting vector selected was in fact not relevant.

Results show that most weights are almost zero and that only a few of them are significantly greater than zero. This is analogous to the procedure adopted by the physician, who only considers a few number of attributes. Therefore, the method is appropriate for selecting relevant attributes and for disregarding the other ones. Table 2 displays the number of relevant weights for all the cases. Two thresholds have been used in this table for considering an attribute to be relevant: 0.1 and 0.05. The table shows that the number of relevant attributes found are similar with normalized and non-normalized data.

A detailed analysis of the relevant attributes shows that, for several cases, the set of relevant attributes is similar in both CBR approaches, the numerical fuzzy similarity approach and the LID method. For example, LID considers the attribute *smoke* as relevant for 12 cases in the case base. For 8 out of these 12 cases the corresponding weighting vectors assign relevant weights to $w_{cb}(smoke)$. Table 3 reviews these cases and provides the relative position of the corresponding weights in the weighting vectors obtained for the same cases. Similarly, LID finds as relevant the attributes $Chol - total$ (10 cases), $HbA1c$ (4 cases), TG (1 case) and $Edu - member$ (1 case). These attributes have also non-zero values in the corresponding weights (5 cases for the $Chol - total$, 4 for $HbA1c$, 1 case for both the TG and $Edu - member$).

The study also shows that there are some cases when this correspondence between attributes in both approaches is not straightforward. Nevertheless, some

Table 2. Number of relevant weights considering two thresholds (0.05 and 0.1): non normalized data (top) and normalized data (bottom). Each column corresponds to one case from CB.

Threshold	Number of relevant weights $w_{cb}(a_i)$ for all 30 cases in CB
0.05	9 7 7 6 7 8 0 7 8 7 6 7 8 8 8 8 6 8 8 8 4 8 8 7 6 8 6 8 8 8
0.1	2 3 4 4 2 3 0 2 3 5 2 1 1 3 3 2 3 2 3 3 3 2 2 3 3 3 3 1 2 2
0.05	7 7 7 7 7 0 0 7 6 8 6 6 7 8 7 8 5 7 7 5 4 8 7 6 6 7 6 7 8 8
0.1	2 3 4 4 4 0 0 2 3 3 1 2 2 3 3 3 4 2 2 3 3 2 2 3 3 3 3 3 3 2

Table 3. Cases where attribute “smoke” was relevant according to the LID method and the relative position of the corresponding weight in the weighting vectors obtained for the same cases. (n.r. stands for “not relevant” weight - i.e., too small weight).

Case where smoke was relevant	Relative position of the weight in normalized data	Relative position of the weight in non-normalized data
1	4-th	3-rd
2	4-th	4-th
6	4-th	n.r.
7	n.r.	n.r.
9	n.r.	n.r.
16	4-th	4-th
17	4-th	4-th
18	4-th	4-th
19	n.r.	n.r.
26	5-th	n.r.

Table 4. Assignments obtained in the classification problem.

	Correct class	Nearest class	Other classes
No normalization	13	16	1
Normalization	13	17	0

of the divergences between both methods are due to the presence of similar or related attributes in both descriptions. For example, the attribute $food-education$ is considered related with another attribute that informs whether the patient goes on some diet or belongs to an educational association.

Additionally, to further evaluate the fuzzy similarity CBR approach, we have considered the classification problem using the previous “diabetes” case base CB. The evaluation has been based on the *leave one out* approach. This is, each time we have removed one case from the case–base and we have computed the similarity of this case with all the remaining cases in the case–base. Table 4 displays the number of correct classifications using this approach. As it can be seen, a little less than half of the cases were correctly classified and for the mis–classified ones, the method assigned the nearest solution class. Here, the nearest solution class is understood in terms of the similarity S_{cl} on the set $\mathbf{R}$ of possible classes. Although the totally correct results in the classification process are not as good as expected, they are indeed quite acceptable as soon as we take into account that, as a matter of fact, we introduced a similarity relation on the solution classes, so not all classes different from the correct one are equally bad. In terms of the plausibility orderings mentioned in Section 2, Table 4 shows that the one of two more plausible solutions is (almost) always is the correct one.

From the case base perspective, the difficulty in correctly classifying the cases can be explained in terms of a lack of redundancy in the case base. It is important to remember that the case–base consisted of only 30 cases in a 84-dimensional space (84 attributes). Therefore, the space of possible problem descriptions is very large, and according to the results, most cases are highly unique in the case–base. In fact, LID yields similar results when evaluated using also the *leave one out* approach. For the same case base, 4 cases were incorrectly classified, while for the rest of the cases LID always produced two answers, the correct class and one of its nearest classes.

6 Conclusions and Future Work

In this paper, within a fuzzy CBR model, we have described a method to construct a particular type of global similarity measure in the problem description domain. Namely, given a case base CB, the similarity between a problem description d and a case $c \in CB$ is defined in terms of a weighted average of the attribute similarities, where the weights are particular to each c. We have applied the approach in a medical domain and compared with another case–based reasoning method, LID, which uses a relational approach with feature terms as a kind of symbolic similarity between cases. The comparison results shows the suitability of the approach.

In particular, results show that only a few attributes are relevant for each case (this is analogous to the procedure adopted by physicians) and that these attributes correspond, in general, to the ones selected by the alternative case–based reasoning method. Classification performances of both approaches also lead to similar results.

Future extensions of this work include the study of additional types of global similarity measures (e.g., using other aggregation operators instead of the weighted mean in (1)). The consideration of non–linear models for defining the similarity relations relate this work with Kernel functions and non–linear Support Vector Machines [34] that, as it is known, is a classification method that operates by mapping data into an alternative and higher–dimensional space so that classes are linearly separable. In our case, the consideration of such transformations would probably improve the performance of the system when still using weighted average based similarity. This process would be an analogue to the one described in [27] for using the Fuzzy C-Means clustering method in the new high–dimensional space.

Acknowledgments

The authors are partially supported by the EU project IBROW (IST-1999-2005) and the CICYT projects STREAMOBILE (TIC2001-0633-C03-02) and e-INSTITUTOR (TIC2000-1414). The authors thank Dr. Albert Palaudàries for his assistance in developing the diabetes application.

References

1. Aamodt, A., Plaza, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* **7**, 1994, pp. 39–59.
2. Aït-Kaci, H., Podelski, A. Towards a meaning of LIFE. *J. Logic Programming* **16**, 1993, pp. 195–234.
3. Armengol, E., Palaudaries, A., and Plaza, E. Individual prognosis of diabetes long–term risks: A CBR approach. *Methods of Information in Medicine* (special issue on prognosis models in medicine: AI and Statistics) **40**, 2001, pp. 46–51.
4. Armengol, E., Plaza, E. Bottom-up induction of feature terms. *Machine Learning* **41**, 2002, pp. 259–294.
5. Armengol, E., Plaza, E. Lazy induction of descriptions for relational case-based learning,in: *Machine Learning ECML-2001*, *Lecture Notes in Artificial Intelligence* **2167**, Springer, 2001, pp. 13-24.
6. Bonissone, P., Ayub, S. Representing cases and rules in plausible reasoning systems, in: *Proceedings of the 1994 ARPA/RL Planning Initiative*, Tucson, AZ,USA, 1994, pp. 305–316.
7. Bonissone P., Cheetman W. (1997). Applications of fuzzy case–based reasoning to residential property valuation, in: *Proceedings of the 6th IEEE Int. Conference on Fuzzy Systems* FUZZ-IEEE'97, Barcelona, Spain, 1997, pp. 37–44.
8. Bonissone, P., López de Mántaras. *Fuzzy Case–Based Reasoning Systems*, in: *Handbook of Fuzzy Computing* (Ruspini, E., Bonissone, P. P., and Pedrycz, W. (eds.)). IOS Publishing Ltd., Amsterdam, F 4.3: pp. 1–17.
9. Burkhard, H.-D., Richter, M. On the notion of similarity of case–based reasoning and fuzzy theory, in: *Soft Computing in Case Base Reasoning* (S. K. Pal, T. S. Dillon, and D. S. Yeung (eds.)). Springer, Berlin, 2001, pp. 29–46.
10. de Calmès, M., Dubois, D., Hüllermeier, E., Prade, H., and Sèdes, F. Case–based querying and prediction: A fuzzy set approach, in: *Proc of the IEEE International Conference on Fuzzy Systems*, Hawaii, USA, 2002, pp. 735–740.
11. Carpenter, B. *The Logic of Typed Feature Structures.* Tracts in Theoretical Computer Science. Cambridge Univ. Press, Cambridge, 1992.
12. Cheetham, B., Cuddihy, P., and Goebel, K. Applications of soft CBR at General Electric,in: *Soft Computing in Case Base Reasoning* (S. K. Pal, T. S. Dillon, and D. S. Yeung (eds.)). Springer, Berlin, 2001, pp. 335-365.
13. Dubois, D., Esteva, F., Garcia, P., Godo, L., Lòpez de Màntaras, R., and Prade, H. Fuzzy modelling of case–based reasoning and decision, in: *Proceedings of the 2nd International Conference on Case–Based Reasoning* ICCBR-97. *Lecture Notes in Artificial Intelligence* **1266**, 1997, pp. 599–610.
14. Dubois, D., Esteva, F., Garcia, P., Godo, L., Lòpez de Mantaras, R., and Prade, H. Fuzzy set modeling in case-based reasoning. *International Journal of Intelligent Systems* **13:4**, 1998, pp. 345–373.
15. Dubois D., Hüllermeier E., and Prade H. (2000), Formalizing case–based inference using fuzzy rules, in: *Soft Computing in Case–Based Reasoning* (S.K. Pal, D.Y. So, and T. Dillon, (eds.)). Springer, Berlin, 2000, pp. 47–72.
16. Dubois D., Hüllermeier E., and Prade H. (2000), Flexible control of case–based prediction in the framework of possibility theory, in: *Proceedings of EWCBR'2000. Lecture Notes in Computer Science*, 1998, pp. 61–73.

17. Dubois D., Prade H. What are fuzzy rules and how to use them.*Fuzzy Sets and Systems* **84**, 1996, pp. 169–185.
18. Esteva, F., Garcia, P., and Godo, L. Fuzzy similarity–based models in case–based reasoning, in: *Proceedings of the 11th IEEE International Conference on Fuzzy Systems* FUZZ-IEEE 2002, Hawaii, USA, 2002, pp. 1348–1353.
19. Filev. D. P., Yager, R. R. On the issue of obtaining OWA operator weights. *Fuzzy Sets and Systems* **94**, 1998, pp. 157–169.
20. Grabisch, M., Nguyen, H. T., and Walker, E. A. *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*. Kluwer, 1995.
21. Hansem, B., Riordan, D. Fuzzy case–based prediction of ceiling and visibility,in: *Proceedings of the 1st Conference on Artificial Intelligence*. American Metereological Society, 1998, pp. 118–123.
22. Hüllermeier, E., Dubois, D., and Prade, H. Knowledge based extrapolation of cases: a possibilistic approach, in: *Proceedings IPMU'2000*, Madrid, Spain, 2000, pp. 1575–1582.
23. Jaczynski M., Trousse B. Fuzzy logic for the retrieval step of a case–based reasoner, in: *Proceedings of the EWCBR'94*, 1994, 313–321.
24. Kolodner J. *Case-Based Reasoning*. Morgan Kaufmann Publishers, 1993.
25. López de Mántaras, R. A distance-based attribute selection measure for decision tree induction. *Machine Learning* **6**, 1991, pp. 81–92.
26. Marichal, J.-L., Roubens, M. Determination of weights of interacting criteria from a reference set. Papiers de Recherche, Faculté d'Economie de Gestion et de Sciences Sociales, Groupe d'Etude des Mathematiques du Management et de l'Economie, 1999, N. 9909.
27. Miyamoto, S., Suizu, D. Fuzzy c-Means clustering using transformations into high dimensional spaces, in: *Proceedings SCIS&ISIS* (CD-ROM), Tsukuba, Japan, 2002.
28. Muggleton, S., De Raedt, L. Inductive logic programming: Theory and methods. *Journal of Logic Programming* **19-20**, 1994, pp. 629–679.
29. Luenberger, D. G. *Introduction to Linear and Nonlinear Programming*. Addison–Wesley, 1973.
30. Plaza, E., Esteva, F., Garcia, P., Godo, L., and López de Màntaras, R. A logical approach to case–based reasoning using fuzzy similarity relations. *Information Sciences* **106**, 1998, pp. 105–122.
31. Tanaka, A., Murofushi, T. A learning model using fuzzy measure and the Choquet integral,in: *Proceedings of the 5th Fuzzy System Symposium*, Kobe, Japan, 1989, pp. 213–217 (in Japanese).
32. Torra, V. On the learning of weights in some aggregation operators: the weighted mean and OWA operators. *Mathematics and Soft Computing* **6**, 2000, pp. 249–265.
33. Torra, V. Learning weights for the quasi–weighted means. *IEEE Transactions on Fuzzy Systems* **10:5**, 2002, pp. 653–666.
34. Vapnik, V. N. *The Nature of the Statistical Learning Theory* 2nd ed., Springer, New York, 2000.
35. Yager R. R. Case–based reasoning, fuzzy systems modelling and solution composition, in: *Proceedings of ICCBR-97*, 1997, pp. 633–643.
36. Yager, R. R. On ordered weighted averaging aggregation operators in multi–criteria decision making. *IEEE Transactions on SMC* **18**, 1998, pp. 183–190.
37. Zadeh L. A. (1971). Similarity relations and fuzzy orderings. *Journal of Information Sciences*, 1971, 177–200.

Incremental versus Non-incremental Rule Induction for Multicriteria Classification

Salvatore Greco[2], Roman Słowiński[1],
Jerzy Stefanowski[1,⋆], and Marcin Żurawski[1]

[1] Institute of Computing Science, Poznań University of Technology,
Piotrowo 3A, 60-965 Poznań, Poland
jerzy.stefanowski@cs.put.poznan.pl, slowinsk@put.poznan.pl,
mzurawski@mikrob.pl

[2] Faculty of Economics, University of Catania, 95129 Catania, Italy
salgreco@mbox.unict.it

Abstract. Induction of decision rules within the dominance–based rough set approach to the multicriteria and multiattribute classification is considered. Within this framework, we discuss two algorithms: *Glance* and an extended version of *AllRules*. The important characteristics of *Glance* is that it induces the set of all dominance–based rules in an incremental way. On the other hand, *AllRules* induces in a non–incremental way the set of all robust rules, i.e. based on objects from the set of learning examples. The main aim of this study is to compare both these algorithms. We experimentally evaluate them on several data sets. The results show that *Glance* and *AllRules* are complementary algorithms. The first one works very efficiently on data sets described by a low number of condition attributes and a high number of objects. The other one, conversely, works well on data sets characterized by a high number of attributes and a low number of objects.

Keywords: rule induction, incremental learning, multiple criteria decision analysis, classification and sorting

1 Introduction

Learning process is sometimes performed in an incremental way. In particular, it refers to human learning. People learn concept descriptions from facts and incrementally refine those descriptions when new facts become available. Newly gained information is rather used to refine knowledge structures and rarely causes reformulation of all knowledge a person has about the subject at hand. Michalski has distinguished in [20] two main reasons why human learn incrementally:

– Sequential flow of information. A human typically receives information in portions and must learn to deal with a situation long before all information about it is available. When new information does become available, there is rarely time to reformulate everything known about all the involved concepts.

⋆ The corresponding author.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 33–53, 2004.

– Limited memory and processing power. People cannot store and have easy access to all the information they have been exposed to. They seem to store only the most prominent facts and generalizations, then modify the generalizations when new facts are available.

These considerations are valid not only for analyzing human learning but also apply to machine learning. An incremental construction of knowledge from examples may be sometimes desirable but it is also indispensable in other cases, see, e.g., discussions in [2, 16, 20]. There are two situations, when it becomes important:

– Temporal–batch learning, when examples are periodically available to the learning system in some portions (subsets) and it is not acceptable to induce knowledge again from complete data, e.g., because of computational costs or inaccessibility of previously processed examples.
– The data source has a dynamic character and provides learning examples in a sequential way, e.g. as series of observations or measurements; it is desirable to update the knowledge representation after reading each example.

It should also be noticed that incremental mining is perceived as an important aspect of knowledge discovery in large databases. For instance, Han claims in [13, p. 220] that "Given the huge amount of data in a database, it is highly preferable to update data mining results *incrementally* rather than mining from scratch on each database update. Thus incremental data mining is an attractive goal for many kinds of mining in large databases or data warehouses". Moreover, Michalski [18] and Xindong Wu [34] have considered incremental learning as one of the most important and challenging research frontiers in inductive learning. Xindong Wu has also paid attention to the following questions concerning these kinds of learning algorithms: When a set of learning examples is large, how can they speed up their learning process? When a new example is input, is it necessary to produce a new rule or revise an existing rule? When some inconsistency is found in a set of examples or knowledge representation just produced, how can they remove it?

Although the majority of existing machine learning algorithms works in a non–incremental way, there is a growing number of solutions employing incremental strategies. The AQ–like family of algorithms for rule induction has its incremental versions, e.g., AQ11 [19], AQ15 [15], or GEM [20]. This learning paradigm is also present in extension matrix approaches as AE5 [14]. Cichosz discusses in [2] how to modify a popular rule induction algorithm CN2 to process incrementally a training set. There are several proposals for decision tree generation, e.g., ID4 [3], ID5R [32], or windowing technique in C4.5 [22] can be viewed as good examples of research on incremental learning. Furthermore, let us notice that several algorithms for learning neural networks, e.g., backpropagation, process the input examples just in an incremental way. For more extensive overview of these methods the reader is referred to, e.g., [16, 21]. This kind of learning has already been considered within the framework of rough sets theory, e.g., Shan and Ziarko [30] introduced a special incremental algorithm for all reduct based rules, Susmaga discussed and experimentally compared incremental methods for reduct generation [31], and Wojna presented [33] a simple incremental modification of a procedure for maintaining the set of reduct rules.

In this paper, we are going to consider incremental induction of decision rules in the context of *Multiple–Criteria Decision Analysis* (shortly abbreviated as MCDA) [4, 24]. More precisely, we will consider one of the major MCDA problems called *multiple–criteria classification* (or sorting) *problem*. It concerns an assignment of objects (representing decision alternatives), evaluated by a set of *criteria*, into pre–defined and preference–ordered decision classes. Criteria are not just regular attributes (typically considered in knowledge discovery or machine learning) but they are attributes with *preference ordered domains*. Handling this specific semantic information on preference orders in data requires new methods. Therefore, Greco, Matarazzo and Slowinski [4] have proposed an extension of rough set theory [23], which is called *Dominance–based Rough Set Approach* (shortly, DRSA). Decision rules, which are induced from learning examples within DRSA framework, have a special syntax that requires a new type of algorithms for their induction.

In previous research, we have introduced an algorithm *DomLem* [7], which allows inducing a *minimal set* of dominance–based rules covering all examples in the input data. On the other hand, sometimes it is also interesting to induce other sets of rules, for example a set of all rules which can be induced from the given data or a subset of all rules which satisfy user predefined requirements concerning, e.g., a minimal number of supporting objects or a maximal length of the condition part. Two algorithms for inducing such sets of rules, called *AllRules* and *DomExplore*, have already been discussed in [27]. The *AllRules* algorithm is strongly tied to multicriteria problems specificity on account of using particular properties of objects belonging to rough approximations. It induces the set of *all* object–based, so called *robust*, rules. It means that the syntax of such robust rule is constructed using a reduced description of an object being the basis for this rule. However, these algorithms work non–incrementally, i.e., all examples need to be read into operating memory before induction. In particular, it makes their application in processing of large databases very difficult, as it would be too expensive to store all data in memory or to scan a database many times.

Therefore, in this paper we discuss an algorithm, called *Glance*, for inducing dominance based rules in an incremental way. It has been first introduced in [35, 29] and handles both regular attributes and criteria describing sorting problems. Moreover, we will present an extended version of *AllRules* algorithm, which should have better scalability and lower computational costs than the previous one. The main aim of this paper is to compare these both algorithms. We experimentally evaluate *Glance* and *AllRules* on several data sets taking into account time of computation, scalability considering number of objects, attributes and different degree of inconsistency.

The paper is organized as follows. The following section contains brief reminder of basic concepts of dominance–based rough set approach and decision rules. In the next sections we describe thoroughly both of rule induction algorithms *Glance* and *AllRules*. In Sect. 5, we describe our experimental design, data sets used for the evaluation, and we also present obtained results. After analyzing these results we conclude with final remarks.

2 Multiple Criteria Classification and Decision Rules

2.1 Preliminaries

When a knowledge representation is induced from learning examples representing multiple criteria decision problems one has to take into consideration semantic information about *preference order* in domains of criteria. For example, such an attribute like “investment cost” has a clear preference scale – the less the better, so it is a criterion. When several criteria are present in a data table, any reasonable regularities to be discovered have to take into account a *semantic correlation* between criteria [4, 9]. For example, in decision about credit granting, where two criteria are considered, “month salary” and “evaluation of bank risk concerning payment of a credit”, these criteria are semantically correlated in the following sense: an improvement of month salary should not deteriorate the evaluation of bank risk. This kind of semantic information is often present in data related to cost or quality, like in financial data, however, it is neglected by today’s data mining tools.

In *multiple criteria classification problems*, criteria describing objects are semantically correlated with preference ordered decision classes. So, while solving such problems, one expects that, e.g., an object x having better, or at least the same, evaluations on a set of criteria than object y cannot be assigned to a worse class than object y. Some data sets may contain objects violating this principle. This constitutes a kind of *inconsistency* that should be discovered in order to avoid wrong conclusions, i.e., decision rules, from MCDA point of view. For this reason, Greco, Matarazzo and Slowinski [4–6] have proposed an extension of the rough set theory that is able to deal with this kind of inconsistency typical to exemplary decisions in MCDA problems. This innovation is mainly based on substitution of the *indiscernibility relation* by a *dominance relation* in the rough approximations of decision classes. In this paper, selected concepts of Dominance-based Rough Set Approach (DRSA) are presented; for more details the reader is referred, e.g., to [4–6].

2.2 Dominance Based Rules

Let us assume that learning examples are represented in *decision table* $DT = (U, A \cup \{d\})$, where U is a set of examples (objects), A is a set of *condition attributes* describing examples, V_a is a domain of a. Let $f(x, a)$ denote the value of attribute $a \in A$ taken by object $x \in U$. The domains of attributes may be preference–ordered or not – in the first case the attributes are called *criteria* while in the latter case we call them *regular attributes*. Preference scale of each criterion a induces a *complete preorder* (i.e. a strongly complete and transitive binary relation) $\succeq$ in the domain of criterion $a \in A$. For objects $x, y \in U$, $a(x) \succeq a(y)$ means that object x is "*at least as good*" as y with respect to criterion a. The asymmetric part of $\succeq$ is denoted by $\succ$. In other words, object x is preferred to object y on criterion a if $a(x) \succ a(y)$. The symmetric part of $\succeq$ is denoted by $\sim$. Therefore, $a(x) \sim a(y)$ means that object x is indifferent to object y with respect to criterion a.

$d \notin A$ is a decision attribute that partitions a set of examples into k decision classes $\mathbf{Cl} = \{Cl_t : t = 1, \ldots, k\}$. These classes are preference ordered, i.e. for all r, s such that $s > r$, objects from Cl_s are preferred to objects from Cl_r. In multiple criteria classification problems it is typical to consider *upward unions* and *downward unions* of classes instead of single decision classes. The upward and downward unions are defined, respectively, as:

$$Cl_g^{\geq} = \bigcup_{h \geq g} Cl_h, \;\; Cl_g^{\leq} = \bigcup_{f \leq g} Cl_f, \; g = 1, \ldots k$$

The statement $x \in Cl_g^{\geq}$ means "x belongs to at least class Cl_g", while $x \in Cl_g^{\leq}$ means "x belongs to at most class Cl_g".

We denote by $A^{\succeq}$ the subset of attributes being criteria in A and by $A^{=}$ the subset of regular attributes, such that $A^{\succeq} \cup A^{=} = A$ and $A^{\succeq} \cap A^{=} = \emptyset$. For each regular attribute $a \in A^{=}$, an *indiscernibility relation*, denoted by $=$, is defined, such that for each $x, y \in U$ $a(x) = a(y)$ means that x is indiscernible with y with respect to the regular attribute a. Moreover, for each $P \subseteq A$ we denote by $P^{\succeq}$ the subset of criteria contained in P, $P^{\succeq} = P \cap A^{\succeq}$, and by $P^{=}$ the subset of regular attributes contained in P, i.e. $P^{=} = P \cap A^{=}$.

We want to approximate upward and downward unions of decision classes from $\mathbf{Cl}$ with respect to $P \subseteq A$ using the binary relation R_P, defined as follows: for each $x, y \in U$ xR_Py if $a(x) \succeq a(y)$ for each $a \in P^{\succeq}$ and $a(x) = a(y)$ for each $a \in P^{=}$.

For any subset $P \subseteq A$ and for any $x \in U$ two sets of objects $R_P^+(x)$ and $R_P^-(x)$ are defined as follows:

$$R_P^+(x) = \{y \in U : yR_Px\}, \;\; R_P^-(x) = \{y \in U : xR_Py\}.$$

In other words, $R_P^+(x)$ is the set of all objects $y \in U$ which dominate x with respect to $P^{\succeq}$ (criteria of P) and are indiscernible with x with respect to $P^{=}$ (attributes of P). Analogously, $R_P^-(x)$ is the set of all objects $y \in U$ which are dominated by x with respect to $P^{\succeq}$ and are indiscernible with x with respect to $P^{=}$.

The sets $R_P^+(x)$ and $R_P^-(x)$ are used to define lower and upper approximations of upward unions $Cl_g^{\geq}$, denoted by $\underline{P}(Cl_g^{\geq})$ and $\overline{P}(Cl_g^{\geq})$, respectively, as follows:

$$\begin{aligned} \underline{P}(Cl_g^{\geq})) &= \{x \in U : R_P^+(x) \subseteq Cl_q^{\geq}\}, \\ \overline{P}(Cl_g^{\geq})) &= \{x \in U : R_P^-(x) \cap Cl_q^{\geq} \neq \emptyset\}. \end{aligned}$$

Analogously, lower and upper approximations of downward unions $Cl_g^{\leq}$, denoted by $\underline{P}(Cl_g^{\leq})$ and $\overline{P}(Cl_g^{\leq})$, respectively, are defined as follows:

$$\underline{P}(Cl_g^{\leq})) = \{x \in U : R_P^-(x) \subseteq Cl_q^{\leq}\},$$
$$\overline{P}(Cl_g^{\leq})) = \{x \in U : R_P^+(x) \cap Cl_q^{\leq} \neq \emptyset\}.$$

The lower approximation of $Cl_g^{\geq}, g \leq k$ contains all objects belonging to $Cl_q^{\geq}$ without any ambiguity with respect to their descriptions. The upper approximation of $Cl_g^{\geq}$ contains all objects which possibly belong to $Cl_q^{\geq}$, i.e., taking into account possible ambiguity in their description. Rough approximations of downward unions of classes admit an analogous interpretation.

Learning examples belonging to rough approximations of upward and downward unions of classes are used for induction of "$if \ldots then \ldots$" decision rules. The two kinds of these rules are distinguished:

$D_{\succeq}$ –decision rules with the following syntax: *if* $(f(x,a_1) \succeq r_{a_1}) \wedge \ldots \wedge (f(x,a_m) \succeq r_{a_m}) \wedge f(x,a_{m+1}) = r_{a_{m+1}}) \wedge \ldots \wedge (f(x,a_p) = r_{a_p})$ *then* $x \in Cl_g^{\geq}$,

$D_{\preceq}$ –decision rules with the following syntax: *if* $(f(x,a_1) \preceq r_{a_1}) \wedge \ldots \wedge (f(x,a_m) \preceq r_{a_m}) \wedge f(x,a_{m+1}) = r_{a_{m+1}}) \wedge \ldots \wedge (f(x,a_p) = r_{a_p})$ *then* $x \in Cl_g^{\leq}$,

where $P = \{a_1, \ldots, a_p\} \subseteq A, P^{\succeq} = \{a_1, \ldots, a_m\}, P^{=} = \{a_{m+1}, \ldots, a_p\}$, $(r_1, \ldots, r_p) \in V_{a1} \times V_{a2} \times V_{ap}$ and $g = 1, \ldots k$.

The $D_{\succeq}$–rule says: "if an evaluation of object x on criterion a_i is at least as good as a threshold value r_i $(i = 1, \ldots, m)$ and on attribute a_j object x is indiscernible with value r_j $(i = m+1, \ldots, p)$ then object x belongs to a least class Cl_g". Similarly, the $D_{\preceq}$–decision rule means that an object is evaluated as "at most as good as a value" and belongs at most to a given class. If the rules are induced from examples belonging to lower approximations of unions, they are called *certain* rules because they assign objects to unions of classes without ambiguity. On the other hand, the *possible* rules are induced from examples belonging to upper approximations. These rules indicate that an object could belong to "at least class Cl_g" or "at most class Cl_g", respectively. Distinction between certain and possible rules is typical for rough set theory [10, 27]. Yet another option is to induce one type of rules only, using Variable Consistency Dominance based Rough Sets Approach (VC-DRSA) [8, 27].

Each dominance based decision rule has to be *minimal.* Since a decision rule is an implication, by a minimal decision rule we understand such an implication that there is no other implication with an antecedent of at least the same weakness (in other words, rule using a subset of elementary conditions or/and weaker elementary conditions) and a consequent of at least the same strength (in other words, rule assigning objects to the same union or sub-union of classes).

Consider a $D_{\succeq}$–decision rule *if* $(f(x,a_1) \succeq r_{a_1}) \wedge \ldots \wedge (f(x,a_m) \succeq r_{a_m}) \wedge f(x,a_{m+1}) = r_{a_{m+1}}) \wedge \ldots \wedge (f(x,a_p) \sim r_{a_p})$ *then* $x \in Cl_g^{\geq}$. If there exists an object y such that $f(y,a_1) = r_{a_1} \wedge \ldots \wedge (f(y,a_m) = r_{a_m} \wedge \ldots \wedge f(y,a_p) = r_{a_p}$, then y is called basis of the rule. Each $D_{\succeq}$–rule having a basis is called *robust* because it is "founded" on an object existing in the data table. Analogous definition of robust decision rules holds for the other types of rules.

We say that an object *supports* a decision rule if it matches both condition and decision parts of the rule. On the other hand, an object is *covered* by a decision rule if it matches the condition part of the rule. The rule can be characterized by a parameter called *strength*, which is the number of supporting objects. Variable consistency rules are also characterized by a confidence parameter that controls the discrimination ability of rules [8].

2.3 Induction of Various Sets of Rules

The set of induced rules is *complete*, when it covers all objects from the data table in such a way that objects belonging to lower approximations of unions are re–assigned to their original classes while inconsistent objects are assigned to a cluster of classes referring to their inconsistency [4]. The set of rules is called *minimal* if it is a set of minimal rules that is complete and non–redundant, i.e. exclusion of any rule from it makes it incomplete.

A minimal set of rules can be induced from examples of multicriteria and multiattribute classification problem by means of the *DOMLEM* algorithm [7, 27]. Such set of rules is usually sufficient for aims of *predicting classification* of new (or testing) objects. However, if the aim of the induction process is *descriptive*, i.e., discovered rules should help in explaining or better understanding of circumstances in which decisions were made, other kinds of rule sets are also useful [27]. One possibility is to create *all decision rules*, which could be generated from the given decision table, while another option is to discover the subset of all rules, which satisfy user's requirements, e.g., sufficiently high strength and confidence, or with a limited number of elementary conditions. The algorithms for inducing such sets of dominance–based rules were already discussed in [27, 35]. Let us notice however, that their computational costs are higher than for DOMLEM; in particular, it is true in case of looking for all rules, which is a problem characterized by an exponential complexity with respect to a number of attributes. Moreover, these algorithms require all examples to be read into memory before induction. Let us also remind that some learning problems are characterized by an incremental processing of information. Further, we will present an *incremental learning* algorithm.

3 "Glance" Algorithm

The algorithm is incremental and stores in memory only rules and not descriptions of learning examples. In its basic version, it induces the set of all rules from examples of multiattribute and multicriteria classification problems. These rules are not necessarily based on some particular objects, i.e., they are non–robust. The general scheme of the algorithm for the case of inducing certain rules is presented below.

```
Procedure Glance
(input $U$ -set of examples; $P$ – set of criteria and regular attributes;
$Cl^{\geq\leq} = \{Cl_1^{\leq}, \ldots, Cl_{t-1}^{\leq}, Cl_2^{\geq}, \ldots, Cl_t^{\geq}\}$ – family of unions of decision classes;
output $\mathbf{R}$ – set of decision rules);
begin
    for each union $u \in Cl^{\geq\leq}$ do
    begin
        Let $r$ be a rule with empty condition part;    {it covers each example}
        $R_u \leftarrow r$;    {add to the set of rules for union $u$}
    end{for}
    for each example $x \in U$ do
        for each $u$ such that $f(x,d) \notin u$ do {$d$ is a decision attribute}
            for $r \in R_u$ do
                if ($r$ covers $x$) then    {$r$ does not discriminate properly}
                begin
                    $R_u \leftarrow R_u \setminus r$;
                    for each $a \in P$ do
                    begin    {specialize rules}
                        Let $s$ be a condition on attr. $a$ excluding $x$;
                        $r_{new} \leftarrow r \cup s$;
                        if ($r_{new}$ is minimal) then $R_u \leftarrow R_u \cup r_{new}$;
                    end; {for each $a \in P$}
                end; {if $r$ covers $x$}
    $\mathbf{R} \leftarrow \bigcup_u R_u$;
end{procedure}
```

The main idea of *Glance* is concordant with inductive learning principle, according to which an algorithm starts from the most general description and then updates it (specializes) so that it is consistent with available training data (also considering approximations). The most general description are rules with empty condition part, which means that such rules cover all examples. At the beginning of calculations, a set of rules R_u containing only one rule with an empty condition part is assigned to every union u. When the new object, e.g., $x \in Cl_g$ is available, it is determined for which unions u this object should be treated as a negative example (these unions include all $Cl_f^{\leq}$ such that $f < g$ for downward unions and $Cl_h^{\geq}$ such that $h > g$ for upward unions). For all such unions their rule sets R_u are checked. If any rule $r \in R_u$ covers x, then it has to be updated as it does not discriminate all negative examples properly. It is removed from R_u and specialized by adding on each attribute or criterion an elementary condition that it is not satisfied by description of x. For criteria, elementary conditions are in forms $(f(x,a) \succ v_a)$ or $(f(x,a) \prec v_a)$, depending on the direction of preference and unions; where v_a is the value of criterion a for object x. For regular attributes, conditions are in form $(f(x,a) \notin \{v_1, \ldots, v_l\})$ and v_a is added to this list. After constructing a new rule its minimality has to be checked. While inducing rules incrementally, minimality is checked only between rules concerning the same union of classes. Verification of minimality between unions is performed only at the very end of calculations, when it is certain that no more learning examples will become available. Furthermore, at the end of the induction process, these conditions in all rules are transformed to representation using operators ($\preceq, \succeq, =$) as in the syntax of dominance based rules.

The computational complexity of the basic version of *Glance* is exponential with respect to a number of attributes. The user can define the maximal accepted number of elementary conditions in the syntax of the rule.

There is also an extended version of *Glance* [35], which allows to induce a satisfactory set of rules with other pre–defined constraint expressing minimum accepted (relative) strength of the rule. However, it requires to maintain in memory additional information about the number of processed examples, which are covered by already generated rules satisfying the constraints – for more details see [35].

4 "AllRules" Algorithm

4.1 Basic Version of Algorithm

This algorithm induces the set of all robust rules, which are based on some particular objects. There is no possibility of using user–specified constraints, except maximal length of the rule. Similarly to previous algorithms, *AllRules* can be used for classification problems, using regular attributes and criteria. It works in a non–incremental way. Moreover, it is strongly tied to rough set theory (see Sect. 2) as it uses idea of lower approximation as a basis for inducing rules. The general scheme of *AllRules* algorithm can be described below for *lower approximation* of an *upward union* of decision classes.

```
Procedure AllRules
(input $U$ – set of examples; $P$ – set of criteria and regular attributes;
$Cl_t^{\geq}$ – upward union of decision classes;
output $\mathbf{R}$ – set of decision rules);
begin
    $R \leftarrow \emptyset$;
    $B \leftarrow \emptyset$;     {temporary set of considered attributes}
    repeat
        generate next subset $B \subseteq P$; subsets are generated starting from
        the smallest ones up to the set of all attributes $P$;
        calculate $\underline{B}(Cl_t^{\geq})$;     {calculate lower approximation of $Cl_t^{\geq}$}
        for each object $x \in \underline{B}(Cl_t^{\geq})$ do
        begin
            if ($x$ does not dominate any other $y \in \underline{B}(Cl_t^{\geq})$ and $y \neq x$ ) then
            begin
                create a rule $r$ on the basis of description of object $x$;
                if ($r$ is minimal) then $R \leftarrow R \cup r$
            end   {if $x$ does not dominate}
        end   {for each}
    until ($B = P$)
end{ procedure }
```

AllRules uses breadth–first strategy and that is why it is possible to stop calculations after computing all rules of some desired maximal length. The algorithm works on subsets B of given set of regular attributes and criteria P. It starts from subsets of cardinality equal to 1, then of cardinality equal to 2, and so on, up to the subset equal to the whole set P. For each subset B, the lower approximation of the decision concept is calculated. Next step depends on a direction of union of classes or its approximations that are considered as concept. For instance, in case of upward unions of classes one tries to identify specific objects from the approximation that are located just close to boundary region. In other words, if an object x belonging to lower approximation of the given concept does not dominate any other object y belonging to that approximation, then x is the basis for a rule. Otherwise, the rule based on its description is not minimal. Analogously, for downward unions of classes it is checked for each object x belonging to lower approximation of the concept with respect to B whether that object x is not dominated by any other object y belonging to that approximation; then, x is the basis for a rule. The condition part of the rule r consists of elementary conditions based on all regular attributes or criteria that belong to B. For regular attributes these conditions are in a form $(f(x,a) = v_a)$, where v_a occurs in the object description. Elementary conditions for criteria are in two forms: $(f(x,a) \succeq v_a)$ or $(f(x,a) \preceq v_a)$. The direction of the relation depends on direction of preference order on a criterion and on a direction of the union for which we are inducing rules.

The minimality of rules is tested as follows: the candidate rule r is compared against previously induced rules based on subsets of attributes present in the current subset B. Let us consider the following simple example. The candidate rule r is: *if (mark in Math is at least 8) and (mark in Physics is at least 7) and (mark in Philosophy is at least 7) then (student is at least "medium").* Assume that among previously induced rules there was the following rule r_1: *if (mark in Math is at least 7) and (mark in Physics is at least 7) then (student is at least "medium").* In this case rule r is not minimal because rule r_1 has the same conclusion but weaker premise. Moreover, concerning minimality of generated rules, it is convenient to calculate rules for unions of classes $Cl_t^{\geq}$ started from class k of the best objects and then systematically decreasing t.

4.2 Extended Version of Algorithm

A computational complexity of the *AllRules* algorithm is still exponential. However, as it was noticed in [35] some operations of rule generation process may be unnecessarily repeated several times. Thus a new extended version of this algorithm has been proposed in [35]. The main improvement is based on an observation that every object x that belongs to lower approximation of concept $Cl_g^{\geq}$ calculated with respect to the set of attributes B, belongs also to the lower approximation of this concept calculated with respect to the set of attributes $B^{'}$ which is a superset of B. *AllRules* will generate a rule basing on object x and the set of attributes $B^{'}$, but this rule will not be minimal, as in the set of previously generated rules there exists also a rule based on object x and the smaller set of attributes $B \subseteq B'$. This kind of problem will occur for every superset B' of B, causing unnecessary construction of a large number of rules not minimal. Therefore, it may be profitable to remember for each set of attributes B all the objects covered by the rules based on B and all its subsets, and not to use these objects as basis for new rules for all supersets of B. According to this idea, in the extended version of *AllRules*, for each subset of attributes B_l of size l the set O_l of objects covered by the rules induced basing on objects described by B_l and all its subsets of size $l-1$ is remembered. These objects are not further considered as basis for next rules. This procedure ensures minimality of new rules, so it is not necessary to check it for each new rule. All objects covered by new rules that have been induced are added to O_l. This version of the algorithm should need less amount of computations, however memory requirements for remembering all extra O_l sets may significantly grow with the number of attributes m.

```
Procedure AllRules–Extended
(input $U$ – set of examples; $P$ – set of criteria and regular attributes;
$Cl_t^{\geq}$ – upward union of decision classes; output $\mathbf{R}$ – set of decision rules);
begin
    $R \leftarrow \emptyset$;
    $B \leftarrow \emptyset$;     {temporary set of considered attributes}
    repeat
        generate next subset $B_l \subseteq P$ of size $l$; subsets are generated starting from
        the smallest ones of size 1 up to the set of all attributes $P$,
        each generated set $B_l$ is combined of its subsets $B_{l-1}$ of size $|B_{l-1}| = |B_l| - 1$;
        $O_l \leftarrow \bigcup\{O_{l-1} \ : \ B_{l-1} \subset B_l\}$; { union of objects covered by rules
                            generated basing on all $B_{l-1}$ such that $B_{l-1} \subset B_l$ }
        calculate $\underline{B}_l(Cl_t^{\geq})$;     {calculate lower approximation of $Cl_t^{\geq}$}
        $\underline{B}_l(Cl_t^{\geq} \leftarrow \underline{B}_l(Cl_t^{\geq}) \setminus O_l$; { do not consider objects belonging to $O_l$ }
        for each object $x \in \underline{B}_l(Cl_t^{\geq}$ do
        begin
            if ($x$ does not dominate any other $y \in \underline{B}_l(Cl_t^{\geq}$ and $y \neq x$) then
            begin
                create a rule $r$ on the basis of description of object $x$;
                $R \leftarrow R \cup r$;
                let $O_r$ be a set of objects covered by rule $r$;
                $O_l \leftarrow O_l \cup O_r$ {update set $O_l$}
            end{if $x$ does not dominate}
        end{for each}
    until ($O_l = Cl_t^{\geq}$ for all $B_k$)
end{procedure}
```

5 Experiments

5.1 Design of Experiments

The aim is to compare the performance of the *Glance* algorithm against other algorithms, which can induce the set of all rules for the multicriteria and multiattribute classification problems. These are *AllRules* algorithms in both *basic* and *extended* versions. Additionally, in some experiments we will consider the previously known algorithm *DomExplore*, which works in a non–incremental way and induces the satisfactory set of *non–robust* rules with strength and confidence not less than predefined thresholds. Let us notice, however, that these non–robust rules may differ from ones induced by *Glance*, because they are generated by combining the most promising single conditions referring to descriptions of existing objects – positive examples of the given concept whereas in case of *Glance* elementary conditions of rule are created by an operation of specialization against negative examples. All these algorithms have been implemented by Zurawski [35] in the single software framework, called *4-emka* [17].

In the experiments carried out in this work we want to answer the following questions concerning new and previously known algorithms:

- What is the scalability of these algorithms? How is computational time and number of rules growing with increasing number of objects, attributes and unions of classes?
- How is computational time and number of rules changing considering proportion of regular attributes to criteria? Which type of attributes (with or without preference) is computationally less demanding?
- Is the consistency of data set important for all algorithms? In other words, how does the quality of approximation (expressing number of objects belonging to boundary) affect time of computations?
- How large are differences between considered types of rules (robust – *AllRules*, non-robust in *DomExplore* style or non–robust in *Glance* style)?

5.2 Data Sets

Two kinds of data sets were used for experiments: artificial and real. Data sets belonging to the first type contain random data generated by a computer program based on some assumptions on a typical set of values, called profile, associated with every class represented in data. The data sets were randomly generated according to chosen probability distribution of attribute values (for details see [35]) and differ by the number of objects. Moreover, for each of data sets one could: change the number of attributes while controlling the proportion of criteria to regular attributes; change the number of decision classes; control the required level of inconsistency (except for one experiment, see Table 8, we continued with generating consistent data sets). The artificial data were used mainly for experiments concerning efficiency of algorithms that is time of computations and number of rules produced.

The real–life data sets were obtained either from the repository of Machine Learning databases at UCI [1] or some rough set applications [27], see their characteristics in Table 1. These data were adapted for multicriteria and multiattribute classification problems by specifying the preference order on some previously regular attributes, on the basis of their correlation with the decision attribute (strong negative correlation leads to modeling decreasing preference, while strong positive correlation leads to increasing preference).

Table 1. Characteristics of real data sets used for experiments.

Data		Number of		Quality of
set	Objects	Attributes	Classes	Approximation
Buses	76	8	3	1.00
Iris	150	4	3	0.97
Pima	768	8	2	0.73
Air bricks	216	10	2	0.84
Wine	178	14	3	0.76

Table 2. The computation time (in seconds) for compared algorithms while changing the number of objects.

	No. of	Number of objects							
Algorithm	Attrib.	500	1000	1500	2000	3000	4000	5000	6000
DomExplore	3	0.10	0.33	0.72	1.15	2.69	4.9	8.3	14.99
	6	3.68	21.4	73.7	178.4	704.9	–	–	–
AllRules	3	0.12	0.16	1.41	3.13	7.69	18.4	34.3	53.01
Basic	6	1.43	5.5	13.1	27.2	108.1	281.9	502.9	712.7
AllRules	3	0.10	0.11	0.41	0.99	2.09	5.49	9.55	16.09
Extended	6	0.16	0.55	1.21	2.26	7.14	17.31	28.9	43.22
Glance	3	0.05	0.05	0.05	0.06	0.06	0.06	0.1	0.12
	6	1.26	3.13	5.5	7.86	15.38	21.2	26.31	38.7

If the artificial or real data sets contained inconsistent examples, we generated certain rules only. Let us notice that considered data sets were either consistent or their quality of approximation of the classification was quite high.

5.3 Results

Firstly, all algorithms were compared on the same artificial data sets. In two series of data, with 3 attributes (2 criteria and 1 regular attribute) and 6 (4 criteria and 2 regular attribute), we systematically changed the number of objects from 500 till 6000. The number of decision classes was equal to 3. Results of computation times are presented in Table 2 and numbers of rules in Tables 3, 4 (we do not distinguish between the versions of *AllRules* as they produce the same number of rules). The symbol "–" means that the algorithm exceeded the accepted resources. It should be reminded, that the number of rules induced by algorithms may be different as the *Allrules* algorithm generates robust rules while others induce non-robust ones. All computation times refer to calculations on the same computer system.

Then, in Table 5, we present results of analyzing the influence of changing the number of regular attributes and criteria on the computational time of algorithms, for a fixed number of objects. It is also quite interesting to study variation of the proportion of regular attributes to criteria in the considered data

Table 3. The number of rules induced by different algorithms while changing the number of objects in the data set with 3 condition attributes.

Algorithm	Number of objects					
	1000	2000	3000	4000	5000	6000
DomExplore	27	27	30	35	33	31
AllRules	27	29	32	30	31	29
Glance	35	29	34	39	34	32

Table 4. The number of rules induced by different algorithms while changing the number of objects in the data set with 6 condition attributes.

Algorithm	Number of objects										
	500	1000	1500	2000	2500	3000	3500	4000	4500	5000	6000
DomExplore	331	469	566	599	690	730	–	–	–	–	–
AllRules	211	289	364	416	473	511	572	590	615	645	743
Glance	807	1197	1459	1561	1621	1792	1903	1971	2005	2115	2205

Table 5. The computational time (in seconds) of algorithms while changing the number of regular attributes and criteria in data containing 100 objects.

	No. of criteria and regular attributes					
Algorithm	3	6	9	12	15	18
DomExplore	0	0.11	0.77	9	41.14	284.79
AllRules Basic	0	0.06	0.82	7.58	68.66	624.83
AllRules Extended	0	0.05	0.06	0.38	3.52	29.93
Glance	0	0.22	44.9	439.41	–	–

set. Here, for the data set consisting of 2000 objects described by 6 condition attributes, we prepared 7 versions of different proportions. In case of considering at least two criteria, one of them was defined on real or integer number scale, while the second on an ordered qualitative scale consisting of 6 different terms. In all data sets the number of decision classes was equal to 3. The results are presented in Table 7.

Furthermore, we examined the influence of changing the number of classes. In this case, for a data set with a fixed size we randomly created several versions of the column referring to a decision attribute. The results are summarized in Table 6. In the similar way, we examined the time of computations while changing the quality of approximation (in the sense of the rough set theory) for the compared algorithms – see results in Table 8. The results for *DomExplore* are not reported as for the majority of these data sets with some inconsistency, it exceeded the available memory resources.

Next, we examined the performance of these algorithms on real life data sets. Their results considering the number of rules and also the average rule strength (i.e., average number of learning examples supporting a rule in the

Table 6. The computational time (in seconds) of algorithms while changing the number of classes in data containing 2000 objects described by 6 condition attributes.

	Number of classes				
Algorithm	2	4	6	8	10
DomExplore	213.11	174.6	131.17	128.04	104.68
AllRules Basic	14	38.51	49.54	78.05	77.28
AllRules Extended	1.16	2.82	3.51	5.19	6.83
Glance	1.86	6.43	7.14	8.73	9.78

Table 7. The computation time (in seconds) of algorithms while changing the proportion of regular attributes to criteria in the data sets containing 2000 objects described by 6 condition attributes.

	No. of criteria / reg. attributes						
Algorithm	0 / 6	1 / 5	2 / 4	3 / 3	4 / 2	5 / 1	6 / 0
DomExplore	0.38	4.8	9.1	135.3	334.6	–	–
AllRules Basic	33.8	60.5	54.8	59.0	65.2	59.4	15.3
AllRules Ext.	8.0	8.2	6.4	5.4	4.9	4.9	4.2
Glance	4.1	4.5	4.8	5.7	14.7	103.5	349.6

Table 8. The computational time (in seconds) of algorithms while changing the level of inconsistency (quality of approximation) in the data set containing 500 objects described by 6 condition attributes.

	Quality of approximation					
Algorithm	1.0	0.93	0.83	0.76	0.64	0.56
AllRules Basic	1.54	1.48	1.37	1.26	1.04	0.89
AllRules Extended	0.11	0.22	0.22	0.33	0.33	0.33
Glance	2.74	6.32	7.9	15.1	13.7	13.45

induced rule set) and average rule length (calculated as the number of elementary conditions in the rule) are presented in Table 9. The differences of computation times between algorithms are proportionally similar as for respective artificial data sets, so we skip presenting them.

6 Discussion of Results and Final Remarks

Discovering knowledge from data containing semantic information about preference orders in domains of criteria requires new tools. In this paper, we have considered the dominance–based rough set approach. Within this framework, we discussed two algorithms: *Glance* and an extended version of *AllRules*. Both algorithms induce the set of all rules from provided learning examples of the multicriteria and multiattribute classification problems. However, these sets of all rules obtained by each algorithm are different. Rules produced by the *All-*

Table 9. Characteristic of sets of rules induced from real data sets; Meaning of symbols: (R) – number of rules, (L) – average length of the rule calculated in a number of conditions, (S) – average strength of the rules calculated in a number of supporting learning examples.

Algorithm	DomExplore			AllRules			Glance		
Data set	R	L	S	R	L	S	R	L	S
Buses	332	2.57	25.98	154	1.91	22.88	612	2.96	25.04
Iris	38	1.84	32.1	23	1.48	46.1	66	2.11	34.0
Pima	–	–	–	1660	3.51	20.5	15913	5.14	29.74
Air bricks	–	–	–	227	2.17	41.76	1139	3.77	16.12
Wine	–	–	–	665	2.6	9.92	2767	2.87	7.67

Rules algorithm are robust (i.e. they are based on objects – positive examples of a concept – existing in the decision table), while rules induced by *Glance* are non-robust (i.e., they are induced by means of the specialization operation, which builds elementary conditions to exclude negative examples from the given decision concept). The way of processing input examples constitutes the other difference. Unlike the *AllRules* and other previously known algorithms for the dominance–based rough set approach, the *Glance* works in an incremental way. It means that every time new examples are available it updates so far obtained rules without having to generate new rules from scratch. It tries to use limited operational memory resources as it stores in memory only rules and not processed examples. The extended version of the *AllRules* algorithm is strongly tied to the concept of dominance-based rough approximations and uses their specific properties to reduce the search through the space of object descriptions pretending to be a basis for minimal robust rules.

These algorithms, as well as the previously known simpler version of *AllRules* algorithm and the *DomExplore* algorithm, have been evaluated and compared in many experiments on real and artificial data sets. Below we discuss the main results.

1. Experiments clearly showed that an extended version of the *AllRules* algorithm is always faster than its previous simpler version, see, e.g., respective results in Tables 2, 6, 7. The difference in favor of the extended version tends to be greater while the number of objects is increasing – see, e.g, Table 2, where is it over 16 times faster.
2. Considering changes of the number of objects in the data sets with the fixed number of attributes equal to 3 or 6 (see Table 2) one can notice that *Glance* is the fastest algorithm, *AllRules* is usually the second, and *DomExplore* is the worst (however, for data sets including 4000 objects or more it is even better than *AllRules*). For these data sets, the computation time of *Glance* grows up “linearly” with the number of objects. *AllRules* is the most sensitive to changing the number of objects.

3. The number of attributes is a crucial issue for all compared algorithms – see, e.g., Table 5. It is somehow obvious considering that problem of generating all rules is exponential with respect to the number of attributes. However, we noticed that computation times strongly differ among the compared algorithms. *Glance* is the most sensitive to increasing the number of attributes. This may be explained by analyzing the way it constructs non–robust rules. Notice in other tables that the number of *Glance* rules grows much quicker with the number of attributes than for other algorithms. While learning incrementally, *Glance* has to check all generated rules and specialize some of them when a new example becomes available. On the other hand, *AllRules* algorithm identifies positive examples being basis for robust rules and, due to this constraint, it produces much smaller number of rules. The reader can also notice that *AllRules – Extended* is the fastest when the number of criteria and regular attributes is greater than nine.
4. The difference in numbers of rules induced by the compared algorithms can be analyzed using Tables 3, 4, and 9. The *AllRules* algorithm always generates the smallest number of rules, *DomExplore* is the second, and *Glance* gives the highest number. This result could be expected knowing the difference between the rules generated by each algorithm. The smallest number of rules for *Allrules* is due to its extra constraint that each rule has to be robust, i.e., it has to be based on some particular object. This significantly reduces the number of possible combinations of elementary conditions and, therefore, reduces the list of candidates to be checked for being a rule. On the other hand, *Glance* produces the greatest number of non–robust rules probably because of its specialization technique, which requires the use of many elementary conditions, even if they are not based on any particular object from the concept (it could happen as a result of exclusion of negative examples). *DomExplore* generates a smaller number of non–robust rules as it combines only “the promising” single conditions, which are created using descriptions of objects existing in the data table. Furthermore, robust rules are shorter than non–robust, see Table 9. The highest average strength of rules varies among algorithms depending on the data set. However, the number of rules induced by any of these algorithms seems to be terribly high with respect to the input data set – with exception of data with very small number of attributes presented in Table 3.
5. The results presented in Table 7 indicate that not only the number of attributes but also their type is important for all algorithms. It is clear that for *AllRules* it is better to process rather criteria than regular attributes, while *DomExplore* is more efficient on regular attributes only. This is not surprising knowing ideas behind each algorithm. In particular, let us remind that *DomExplore* is a modified version of the *Explore* algorithm, which was originally designed for handling qualitative regular attributes [27, 28]. *Glance* seems to be located in the middle between both algorithms.
6. Increasing number of decision classes also increases the time of computations. The exception is *DomExplore* for which the time decreases, see Table 6. Nevertheless, it is still the slowest algorithm.

7. The influence of inconsistency in data on the time of computation seems to be different depending on the algorithm. For *DomExplore* and *Glance*, the more data is inconsistent, the harder and more expensive calculations are. In fact, in our experiment *DomExplore* was unable to complete computations for sets with quality of approximation lower than 1 because it exceeded memory resources. Notice in Table 8 that in case of low values of quality, the time of computations by *Glance* may slightly decrease due to low number of rules that can be produced for data with such high inconsistencies. *AllRules* seems to be insensitive to the changes of data consistency.

To sum up, *Glance* and *AllRules* seem to be complementary algorithms. The first one works very efficiently on data sets described by a low number of condition attributes and a high number of objects. The other one inversely – works well on data sets characterized by a high number of attributes and a low number of objects.

The important feature of *Glance* is its ability to learn incrementally. In fact, in our experiments its better performance for data sets characterized by a smaller number of attributes should be even much greater if the examples would be processed really incrementally. Let us assume that examples are available in portions of 500 instances. In such a case, *AllRules* has to repeat all calculations again for all objects available at the moment, while *Glance* only updates the former set of rules according to new examples.

The disadvantage of using *Glance* for a larger number of attributes is mainly due to the enormous number of rules, which leads to high computation times and using too much operating memory for storing already generated rules. The user can overcome this problem partly by using the parameter of maximal length of rules. In an extended version of *Glance* the user can also tune another parameter of minimum relative strength of a rule. Another possibility could be to develop a new version of Glance, which will not generate the set of all rules. For instance, the operation of updating rules, which do not discriminate properly negative examples, could include only a limited number of the best specializations. This is somehow similar to the idea of using constraints on a maximal size of the STAR in *AQ*–like algorithms [18, 19]. As a result, the algorithm should provide much smaller set of "satisfactory" rules. Yet another option could be developing new *Glance* version that would produce incrementally a kind of minimal set of rules. This seems to reduce the use of operating memory and, therefore, be useful for exploring very large databases, as discussed in the introduction. Let us remind here, that the extended version of *AllRules*, although it has been relatively fast in some experiments, is not designed to handle larger number of objects and manage efficiently the size of operating memory. Thus, an incremental induction of robust rules is also a quite challenging task.

Other open research problems could include:

- Evaluating the ability of obtained rule sets to predict classification for new objects. In case of the dominance-based approach this is a different problem than in case of regular attributes, because rules indicate unions of ordered classes instead of single classes. Moreover, using larger set of rules, which is a case in this paper, leads to more frequent conflict situations of matching object descriptions to many rules than while using minimal sets of rules. For more details on such conflicts see, e.g., [11, 12, 27]. The preliminary experiments performed in [35] showed that the accuracy of classification obtained by using sets of all rules induced by considered algorithms is rather low compared to the use of minimal sets induced by *DomLem* [7]. Therefore, it is necessary to develop new classification strategies for determining final decision based on dominance–based rules.

- Comparing more deeply the usefulness of sets of all rules against the minimal set of rules. We can conclude from our current experimental results, that the number of rules induced by any of discussed algorithms is in many cases too high, what is inconvenient for practical problems.
- Adapting some of these algorithms, in particular incremental ones, to handle huge number of learning examples.
- Include the best versions of the algorithms in new *4-emka* or user software systems for public use.

Acknowledgments

The research of the first author has been supported by the Italian Ministry of Education, University and Scientific Research (MIUR). The other authors wish to acknowledge financial support from the State Committee for Scientific Research and by the Foundation for Polish Science.

References

1. Blake, C., Koegh, E., and Mertz, C.J. *Repository of Machine Learning.* University of California at Irvine, 1999.
[URL: http://www.ics.uci.edu/˜mlearn/MLRepositoru.html]
2. Cichosz, P. *Learning systems.* WNT, Warszawa, 2000 (in Polish).
3. Fisher, D., Schlimmer, J. C. A case study of incremental concept induction, in: *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, USA. Morgan Kauffman, 1986, pp. 302–507.
4. Greco, S., Matarazzo, B., and Slowinski, R. The use of rough sets and fuzzy sets in MCDM, in: *Advances in Multiple Criteria Decision Making* (Gal, T., Stewart, T., and Hanne, T. (eds)). Kluwer, 1999, pp. 14.1–14.59.
5. Greco, S., Matarazzo, B., and Slowinski, R. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research* **129: 1**, 2001, pp. 1–47.
6. Greco, S., Matarazzo, B., and Slowinski, R. Multicriteria classification, in: *Handbook of Data Mining and Knowledge Disocvery.* Oxford Univ. Press, 2002, pp. 318–328.
7. Greco, S., Matarazzo, B., Slowinski, R., and Stefanowski, J. An algorithm for induction decision rules consistent with the dominance principle, in: *Rough Sets and Current Trends in Computing. Proceedings of the 2nd Int. Conference RSCTC 2000*, Banff, October, 2000. *Lecture Notes in Artificial Intelligence* **2005**, Springer, 2001, pp. 304–313.

8. Greco, S., Matarazzo, B., Slowinski, R., and Stefanowski, J. Variable consistency model of dominance-based rough set approach, in: *Rough Sets and Current Trends in Computing. Proceedings of the 2nd Int. Conference RSCTC 2000* Banff, October, 2000. *Lecture Notes in Artificial Intelligence* **2005**, Springer, 2001 , pp. 170–181.
9. Greco, S., Slowinski, R., and Stefanowski, J.Mining association rules in preference–ordered data, in: *Foundations of Intelligent Systems. Proceedings of the 13th Int. Symposium ISMIS 2002*, Lyon France, June 2002. *Lecture Notes in Artificial intelligence* **2366**, Springer, 2002, pp. 442–451.
10. Grzymala–Busse, J.W.LERS – a system for learning from examples based on rough sets, in: *Intelligent Decision Support: Handbook of Advances and Applications of Rough Sets Theory*(Slowinski R. (ed.)). Kluwer, 1992, pp. 3–18.
11. Grzymala–Busse, J.W. Managing uncertainty in the machine learning from examples, in: *Proceedings of the 3rd Workshop on Intelligent Information Systems*, Wigry, Poland, IPI PAN Press, 1994, pp. 70–84.
12. Grzymala–Busse, J.W., Zou, X. Classification strategies using certain and possible rules, in: *Rough Sets and Current Trends in Computing. Proceedings of the RSCTC'1998 Conference*, Warszawa, 1998. *Lecture Notes in Artificial Intelligence* **1424**, Springer, 1998, pp. 37–44.
13. Han, J., Kamber, M. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.
14. Hong, J. Learning from examples and a multi–purpose learning system AE5. *Chinese Journal of Computers* **12**, 1989, pp. 98–105.
15. Hong, J. R., Mozetic, I., and Michalski, R. S. *AQ15: Incremental Learning of Attribute–based Descriptions from Examples – the Method and User's Guide.* Report ISG 85-5, UIUCDCS-F-86-949, Dept. of Computer Science, Univ. of Illinois, Urbana, 1986.
16. Langley, P. *Elements of Machine Learning.* Morgan Kaufmann, 1996.
17. Lukasik, M., Luszczynski, M., Szumski, M., and Zurawski, M.. *Multicriteria Decision Aid Software Using Dominance Based Rough Sets and Decision Rules.* Eng. B.Sc. Thesis, Poznan University of Technology, 1999 (in Polish).
18. Michalski, R.S. A theory and methodology of inductive learning, in: *Machine Learning: An Artificial Intelligence Approach* vol. 1 (R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.)). Morgan Kaufman, 1983, pp. 83–134.
19. Michalski, R. S., Larson, J. *Selection of Most Representative Training Examples and Incremental Generation of VL1 Hypothesis: the Underlying Methodology and Description of Programs ESEL and AQ11.*Tech. Report UIUCDCS R 78 867, Dept. of Computer Science, Univ. of Illinois at Champaign-Urbana, Urbana, 1978.
20. Michalski, R. S., Reinke, R. E. Incremental learning of concept descriptions: A method and experimental results, in: *Machine Intelligence* XI. Clarendon Press, 1988, pp. 263–288.
21. Mitchell, T. *Machine Learning.* Mac-Graw Hill, 1997.
22. Quinlan, J. R. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
23. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data.*Kluwer, 1991.
24. Roy, B. *Multicriteria Methodology for Decision Aiding.*Kluwer, 1995.
25. Slowinski, R., Stefanowski, J., Greco, S., and Matarazzo, B. Rough sets processing of inconsistent information. *Control and Cybernetics* **29: 1**, 2000, pp. 379–404.
26. Stefanowski, J. On rough set based approaches to induction of decision rules, in: *Rough Sets in Data Mining and Knowledge Discovery* vol. 1. Physica, 1998, pp. 500–529.

27. Stefanowski, J. (2001) *Algorithims of Rule Induction for Knowledge Discovery.* Habilitation Thesis published as Series Rozprawy no. 361, Poznan Univeristy of Technology Press, 2001, Poznan (in Polish).
28. Stefanowski, J., Vanderpooten, D. Induction of decision rules in classification and discovery–oriented perspectives. *Int. Journal of Intelligent Systems* **16: 1**, 2001, pp. 11–28.
29. Stefanowski, J., Zurawski, M. Incremental learning of decision rules for classification problems. Submitted to IIS-2003, New Trends in Intelligent Information Processing and Webmining Conference, 2002.
30. Shan, N., Ziarko, W. Data–based acquisition and incremental modification of classification rules. *Computational Intelligence* **11: 2**, 1995, pp. 357–370.
31. Susmaga, R. Experiments in incremental computation of reducts, in: *Rough Sets in Data Mining and Knowledge Discovery* vol. 1 (Polkowski L., Skowron, A. (eds.)). Physica, 1998, pp. 530–553.
32. Utgoff, P. E. Incremental induction of decision trees. *Machine Learning* **4**, 1989, pp. 161–186.
33. Wojna, A. Constraint based incremental learning of classification rules, in: *Rough Sets and Current Trends in Computing. Proceedings of the 2nd Int. Conference RSCTC 2000* Banff, October, 2000. *Lecture Notes in Artificial Intelligence* **2005**, 2001, pp. 428–435.
34. Wu Xiandong. Inductive learning: Algorithms and frontiers. *Artificial Intelligence Review* **7**, 1993, pp. 93–108,
35. Zurawski, M. *Algorithms of Induction of Decision Rules for Multicriteria Decision Problems.* M.Sc. Thesis, Poznań University of Technology, 110 pp., 2001.

Three Strategies to Rule Induction from Data with Numerical Attributes

Jerzy W. Grzymala-Busse

University of Kansas, Lawrence, KS 66045, USA

Abstract. Rule induction from data with numerical attributes must be accompanied by discretization. Our main objective was to compare two discretization techniques, both based on cluster analysis, with a new rule induction algorithm called MLEM2, in which discretization is performed simultaneously with rule induction. The MLEM2 algorithm is an extension of the existing LEM2 rule induction algorithm, working correctly only for symbolic attributes and being a part of the LERS data mining system. For the two strategies, based on cluster analysis, rules were induced by the LEM2 algorithm. Our results show that MLEM2 outperformed both strategies based on cluster analysis and LEM2, in terms of complexity (size of rule sets and the total number of conditions) and, more importantly, in terms of error rates.

1 Introduction

Many real-life data contain numerical attributes, with values being integers or real numbers. Such numerical values cannot be used in rules induced by data mining systems since there is a very small chance that these values may match values of unseen, testing cases. There are two possible approaches to processing data with numerical attributes: either to convert numerical attributes into intervals through the process called discretization before rule induction or conduct both discretization and rule induction at the same time.

The former approach is more frequently used in practice of data mining. An entire spectrum of discretization algorithms was invented [7]. Using this approach discretization is performed as a preprocessing for the main process of rule induction.

The latter approach was used in a few systems, e.g., in C4.5 that induces decision trees at the same time discretizing numerical attributes [17], in the MODLEM algorithm [9, 10, 18], a modification of LEM2, and in the MLEM2 algorithm. The MLEM2 algorithm [8] is another extension of the existing LEM2 rule induction algorithm. Performance of MLEM2 was compared with MODLEM performance in [8]. The LEM2 algorithm is a part of data mining system LERS (Learning from Examples based on Rough Sets) [5, 6]. LERS uses rough set theory [14, 15] to deal with inconsistency in input data.

As follows from our previous results on melanoma diagnosis [1], discretization based on cluster analysis is a sound approach. For example, discretization algorithms based on divisive and agglomerative methods were ranked as the second

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 54–62, 2004.

and third (out of six) with respect to error rate (with minimal entropy being first), however, an expert in the domain ranked rule sets induced by LEM2 from the data discretized by minimal entropy on the fourth position, while the discretization algorithms based on divisive method of cluster analysis was again second [1]. In our previous research we used only one data set describing melanoma.

Our current objective was to compare two discretization techniques, based on cluster analysis, with a new rule induction algorithm called MLEM2, in which discretization is performed simultaneously with rule induction. As follows from our results, MLEM2 outperformed two other strategies, in which discretization techniques based on cluster analysis were used first and then rule induction was conducted by LEM2 algorithm. Note that MLEM2 produces the same rule sets from symbolic attributes as LEM2.

2 Discretization Algorithms Based on Cluster Analysis

The data mining system LERS uses for discretization a number of discretization algorithms, including two methods of cluster analysis: agglomerative (bottom-up) [3] and divisive (top-down) [16]. In agglomerative techniques, initially each case is a single cluster, then they are fused together, forming larger and larger clusters. In divisive techniques, initially all cases are grouped in one cluster, then this cluster is gradually divided into smaller and smaller clusters. In both methods, during the first step of discretization, *cluster formation*, cases that exhibit the most similarity are fused into clusters. Once this process is completed, clusters are projected on all attributes to determine initial intervals on the domains of the numerical attributes. During the second step (*merging*) adjacent intervals are merged together. In the sequel, the former method will be called the agglomerative discretization method, the latter will be called the divisive discretization method. In our experiments, both methods used were *polythetic* (*all* numerical attributes were used).

Initially all attributes were categorized into numerical and symbolic. During clustering, symbolic attributes were used only for clustering stopping condition. First, all numerical attributes were normalized (attribute values were divided by the attribute standard deviation, following [4]).

In agglomerative discretization method initial clusters were single cases. Then the distance matrix of all Euclidean distances between pairs of cases was computed. The closest two cases, a and b, compose a new cluster $\{a, b\}$. The distance from $\{a, b\}$ to any remaining case c was computed using the Median Cluster Analysis formula [4],

$$\frac{1}{2}d_{ca} + \frac{1}{2}d_{cb}\frac{1}{4}d_{ab},$$

where d_{xy} id the Euclidean distance between x and y. The closest two cases compose a new cluster, etc.

At any step of clustering process, the clusters form a partition π on the set of all cases. All symbolic attributes define another partition τ on the set of all cases. The set of all concepts define yet another partition λ on the set of all cases. The process of forming new clusters was terminated when $\pi \cdot \tau > \lambda$.

In divisive discretization method, initially all cases were placed in one cluster C_1. Next, for every case the average distance from all other cases was computed. The case with the largest average distance was identified, removed from C_1, and placed in a new cluster C_2. For all remaining cases from C_1 a case c with the largest average distance d_1 from all other cases in C_1 was selected and the average distance d_2 from c to all cases in C_2 was computed. If $d_1 - d_2 > 0$, c was removed from C_1 and put to C_2. Then the next case c with the largest average distance in C_1 was chosen and the same procedure was repeated. The process was terminated when $d_1 - d_2 \leq 0$. The partition defined by C_1 and C_2 was checked whether all cases from C_1 were labeled by the same decision value and, similarly, if all cases from C_2 were labeled by the same decision value (though the label for C_1 might be different than the label for C_2). The stopping condition was the same as for the agglomerative discretization method.

Final clusters were projected into all numerical attributes, defining this way a set of intervals. The next step of discretization was merging these intervals to reduce the number of intervals and, at the same time, preserve consistency. Merging of intervals begins from *safe merging*, where, for each attribute, neighboring intervals labeled by the same decision value are replaced by their union provided that the union was a labeled again by the same decision value. The next step of merging intervals was based on checking every pair of neighboring intervals whether their merging will result in preserving consistency. If so, intervals are merged permanently. If not, they are marked as un-mergeable. Obviously, the order in which pairs of intervals are selected affects the final outcome. In our experiments we started from an attribute with the most intervals first.

3 MLEM2

In general, LERS uses two different approaches to rule induction: one is used in machine learning, the other in knowledge acquisition. In machine learning, or more specifically, in learning from examples (cases), the usual task is to learn discriminant description [13], i.e., to learn the smallest set of minimal rules, describing the concept. To accomplish this goal, i.e., to learn *discriminant description*, LERS uses two algorithms: LEM1 and LEM2 (LEM1 and LEM2 stand for Learning from Examples Module, version 1 and 2, respectively) [5].

Let B be a nonempty lower or upper approximation of a concept represented by a decision-value pair (d, w). Set B *depends* on a set T of attribute-value pairs (a, v) if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq B,$$

where $[(x, v)]$ denotes the set of all examples such that for attribute a its values are v.

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

- each member T of $\mathcal{T}$ is a minimal complex of B;
- $\bigcap_{t \in \mathcal{T}}[T] = B$;
- $\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ has the smallest possible number of members.

The user may select an option of LEM2 with or without taking into account attribute priorities. The procedure LEM2 with attribute priorities is presented below. The option without taking into account priorities differs from the one presented below in the selection of a pair $t \in T(G)$ in the inner loop WHILE. When LEM2 is not to take attribute priorities into account, the first criterion is ignored. In our experiments all attribute priorities were equal to each other.

```
Procedure LEM2
(input: a set $B$,
output: a single local covering $\mathcal{T}$ of set $B$);
begin
    $G := B$;
    $\mathcal{T} := \emptyset$;
    while $G \neq \emptyset$
        begin
        $T := \emptyset$;
        $T(G) := \{t | [t] \cap G \neq \emptyset\}$ ;
        while $T = \emptyset$ or $[T] \nsubseteq B$
            begin
                select a pair $t \in T(G)$ with the highest
                attribute priority, if a tie occurs, select a pair
                $t \in T(G)$ such that $|[t] \cap G|$ is maximum;
                if another tie occurs, select a pair $t \in T(G)$
                with the smallest cardinality of $[t]$;
                if a further tie occurs, select first pair;
                $T := T \cup \{t\}$ ;
                $G := [t] \cap G$ ;
                $T(G) := \{t | [t] \cap G \neq \emptyset\}$;
                $T(G) := T(G) - T$ ;
                end {while}
            for each $t \in T$ do
                if $[T - \{t\}] \subseteq$ B then $T := T - \{t\}$;
            $\mathcal{T} := \mathcal{T} \cup \{T\}$;
            $G := B - \bigcup_{T \in \mathcal{T}}[T]$;
    end {while};
    for each $T \in \mathcal{T}$ do
        if $\bigcup_{S \in \mathcal{T} - \{T\}}[S] = B$ then $\mathcal{T} := \mathcal{T} - \{T\}$;
end {procedure}.
```

For a set X, $|X|$ denotes the cardinality of X.

Rules induced from raw, training data are used for classification of unseen, testing data. The classification system of LERS is a modification of the *bucket brigade algorithm* [2, 12]. The decision to which concept a case belongs is made on the basis of three factors: strength, specificity, and support. They are defined as follows: *Strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of scores of all matching rules from the concept. The concept C for which the support (i.e., the sum of all products of strength and specificity, for all rules matching the case, is the largest is a winner and the case is classified as being a member of C).

MLEM2, a modified version of LEM2, categorizes all attributes into two categories: numerical attributes and symbolic attributes. For numerical attributes MLEM2 computes blocks in a different way than for symbolic attributes. First, it sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint x MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than x, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than x. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Starting from that point, rule induction in MLEM2 is conducted the same way as in LEM2.

4 Experiments

In our experiments we used eight well-known data sets with numerical attributes (Table 1). All of our data sets, except *Bank*, were obtained from the University of California at Irvine Machine Learning Depository. The *Australian Credit Approval* data set was donated by J. R. Quinlan. The data set *Bank* describing bankruptcy was created by E. Altman and M. Heine at the New York University School of Business in 1968. The data set *Bupa*, describing liver disorders, contain data gathered by BUPA Medical Research Ltd., England. *German* data set, with

Table 1. Data sets.

	Number of		
	cases	attributes	concepts
Australian	690	14	2
Bank	66	5	2
Bupa	345	6	2
German	1000	24	2
Glass	214	9	6
Iris	150	4	3
Pima	768	8	2
Segmentation	210	19	7

only numerical attributes, was donated by H. Hoffman from the University of Hamburg (Germany). The data set *Glass*, representing glass types, was created by B. German, Central Research Establishment, Home Office Forensic Science Service, Canada. The *Iris* data set was created by R. A. Fisher and donated by M. Marshall in 1988. The *Pima* data set describes Pima Indian diabetes and was donated by V. Sigillito in 1990. The data set *Segmentation* created in 1990 by the Vision Group, University of Massachusetts, represents image features: brickface, sky, foliage, cement, window, path, and grass.

Table 2 presents means of numbers of errors for all eight data sets. The means of errors were computed using 30 experiments of variable ten-fold cross validation, with the exception of three data sets: Bupa, German and Pima and MLEM2. Only 20 experiments of running of MLEM2 against Bupa and Pima and only 21 experiments of running MLEM2 against German were performed since these experiments were very time-consuming (e.g., running of MLEM2 against Pima lasted for about 2 hours 37 minutes on Alpha 21264 computer). Table 3 presents sample standard deviations of numbers of errors associated with our experiments.

Table 4 presents the cardinalities of rule sets induced by respective methods. Also, Table 5 presents the total numbers of conditions in the corresponding rule sets.

Table 2. Means of numbers of errors.

	Agglomerative Discretization Method	Divisive Discretization Method	MLEM2
Australian	205.5	212.6	135.3
Bank	5.63	4.9	3.73
Bupa	143.8	142.9	109.9
German	288.6	294.0	285.3
Glass	74.1	69.9	71.83
Iris	13.33	9.3	8.97
Pima	253.1	249.7	223.1
Segmentation	50.4	44.9	36.77

Table 3. Standard deviations of numbers of errors.

	Agglomerative Discretization Method	Divisive Discretization Method	MLEM2
Australian	6.23	11.24	7.93
Bank	1.35	1.63	1.11
Bupa	8.87	8.33	6.87
German	7.04	5.97	7.60
Glass	4.25	5.42	4.81
Iris	2.19	1.68	1.27
Pima	8.70	8.74	12.34
Segmentation	5.28	4.53	3.47

Table 4. Sizes of rule sets.

	Agglomerative Discretization Method	Divisive Discretization Method	MLEM2
Australian	115	109	90
Bank	4	6	3
Bupa	164	162	71
German	205	232	159
Glass	82	76	30
Iris	13	11	8
Pima	287	264	116
Segmentation	38	35	14

Table 5. Total numbers of conditions in rule sets.

	Agglomerative Discretization Method	Divisive Discretization Method	MLEM2
Australian	559	542	348
Bank	7	13	6
Bupa	487	500	241
German	1309	1359	814
Glass	262	255	87
Iris	30	21	17
Pima	1025	963	428
Segmentation	144	134	37

5 Conclusions

Our main objective was to compare three different strategies for rule induction from data with numerical attributes. In the first two strategies, data with numerical attributes were discretized first, using two different discretization algorithms, based on agglomerative and divisive algorithms of cluster analysis, respectively, and then rule sets were induced using the LEM2 algorithm. The LEM2 algorithm is the most frequently used rule induction option of the LERS data mining system. In the third strategy we used our new algorithm, called MLEM2, an extension of the LEM2 algorithm. Results of our experiments are presented in Tables 2 and 3. In order to rank these three strategies, first, for any data set and for every pair of means from Table 2 and associated with the three methods, the well known test [11] about the difference between two means, involving error standard deviation and sample sizes was taken with the significance level equal to 0.05. Then, for the three sequences of results of this test, each associated with the eight data sets, the sign test [11]was used to compare the entire methods, again, with the same significance level equal to 0.05. Results are: MLEM2 produces less errors than the remaining two methods and the remaining two methods are incomparable (the null hypothesis that they produce the same number of errors cannot be rejected with the level of significance equal to 0.05).

Less important observation, about the complexity of rule sets, follows from Tables 4 and 5: rule sets induced by MLEM2 are simpler than rule sets induced by the remaining two strategies (the total number of rules and the total number of conditions, for any data sets used in our experiments, were always smaller for MLEM2).

Our final observation is that MLEM2 induces rules from raw data with numerical attributes, without any prior discretization, and that MLEM2 provides the same results as LEM2 for symbolic attributes. Note that MLEM2 can handle also missing attribute values. A comparison of MLEM2 and other approaches to missing attribute values will be reported in the future.

References

1. Bajcar, S., J. W. Grzymala–Busse, and Z. S. Hippe. A comparison of six discretization algorithms used for prediction of melanoma,in: *Proceedings of the Eleventh International Symposium on Intelligent Information Systems*, IIS'2002, Sopot, Poland, June 3–6, 2002, Physica, 2003, pp. 3–12.
2. Booker, L. B., D. E. Goldberg, and J. F. Holland. Classifier systems and genetic algorithms, in: *Machine Learning. Paradigms and Methods* (Carbonell, J. G. (ed.)). MIT Press, 1990, pp. 235–282.
3. Chmielewski, M. R., J. W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *Int. Journal of Approximate Reasoning* **15**, 1990, pp. 319–331.
4. Everitt, B. *Cluster Analysis* 2nd ed.. Heinmann Educational Books, London, 1980.
5. Grzymala-Busse, J. W. (1992) LERS—A system for learning from examples based on rough sets. In Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, Slowinski, R. (ed.), Kluwer Academic Publishers, Dordrecht, Boston, London, 3–18.
6. Grzymala-Busse, J. W. A new version of the rule induction system LERS. *Fundamenta Informaticae* **31** , 1997, pp. 27–39.
7. Grzymala–Busse, J. W. Discretization of numerical attributes, in: *Handbook of Data Mining and Knowledge Discovery* (Klösgen, W., J. Zytkow (eds)). Oxford Univ. Press, 2002, pp. 218–225.
8. Grzymala–Busse, J. W. MLEM2: A new algorithm for rule induction from imperfect data,in: *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU 2002, Annecy, France, July 1–5, 2002, pp. 243–250.
9. Grzymala–Busse, J. W., J. Stefanowski. Discretization of numerical attributes by direct use of the rule induction algorithm LEM2 with interval extension,in: *Proceedings of the Sixth Symposium on Intelligent Information Systems*, IIS'97, Zakopane, Poland, June 9–13, 1997, pp. 149–158.
10. Grzymala–Busse, J. W., J. Stefanowski. Three discretization methods for rule induction. *International Journal of Intelligent Systems* **16**,2001, pp. 29–38.
11. Hamburg, M. *Statistical Analysis for Decision Making.* Harcourt Brace Jovanovich, 1983, pp. 546–550; 721.
12. Holland, J. H., K. J. Holyoak, R. E. Nisbett. *Induction. Processes of Inference, Learning, and Discovery.* MIT Press, 1983.

13. Michalski, R. S. *A Theory and Methodology of Inductive Learning*, in: *Machine Learning. An Artificial Intelligence Approach* (Michalski, R. S., J. G. Carbonell, and T. M. Mitchell (eds.)). Morgan Kauffman, 1983, pp. 83–134.
14. Pawlak, Z. Rough Sets. *International Journal of Computer and Information Sciences* **11**, 1982, pp. 341–356.
15. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, 1991.
16. Peterson, N. Discretization using divisive cluster analysis and selected post–processing techniques. Internal Report, Department of Computer Science, University of Kansas, 1993.
17. Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
18. Stefanowski, J. On rough set based approaches to induction of decision rules, in: *Rough Sets in Data Mining and Knowledge Discovery* (Polkowski, L., A. Skowron (eds.)). Physica, 1998, pp. 500–529.

Fuzzy Transforms

Irina Perfilieva

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22,
701 03 Ostrava 1, Czech Republic

Abstract. The technique of direct and inverse F–transforms is introduced and approximating properties of the inverse F–transform are described. Numerical methods based on fuzzy approximation models are developed. Among them are numeric integration and approximate solution of ordinary differential equations (the Cauchy problem). Advantages of the new technique based on F–transforms over some classical numeric methods are demonstrated on examples.

Keywords: universal approximation, fuzzy transform, basic function

1 Introduction

Fuzzy logic has been demonstrated to provide a basis for the approximate description of different dependencies. Fuzzy logic's ability to produce smooth descriptions has especially attracted researchers from various areas. A lot of results are unified by the common name – the universal approximation, which has proved the approximation property of so called fuzzy models. Generally speaking, each model differs from the other ones by the choice of logical operations.

Unfortunately, the technique of producing fuzzy approximation models has not been followed by other methods. We know from the theory of numerical methods that approximation models can replace original complex functions in some computations, e.g. in solving of differential equations, etc. This chapter is a new contribution to this area. We call this type of technique *numerical methods based on fuzzy approximation models.*

In this chapter, we show how it is possible to obtain an approximate value of a definite integral as well as an approximate solution to an ordinary differential equation on the basis of the certain class of fuzzy approximation models. The way how the parameters of the models are computed is analogous to those used in the theory of generalized functions and Fourier and Laplace transforms. On the basis of this analogy we introduce our technique as a kind of transformation of the given function and call it fuzzy transforms or shortly F–transforms.

2 Preliminaries: Approximation Models Instead of Data

We confine ourselves to approximation models which can be represented by functions of one variable. The generalization to the case of two and more variables is straightforward.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 63–81, 2004.

Suppose that we are given data and are concerned with construction of a representative model. By this we mean the construction of a formula which represents (precisely or approximately) the given data. For example, if our data comprise a collection of pairs (x_i, y_i), $i = 1, \dots, n$, then the model can be given by a formula $F(x)$ such that the function $y = F(x)$ interpolates or approximates the data. In the first case we have,

$$y_i = F(x_i), \quad i = 1, \dots, n,$$

while in the second case, the equality may not hold, but the function $y = F(x)$ "nicely" approximates (in the sense of some criterion) the data. We choose interpolation when our data are precise, otherwise we look for an approximation. Let us stress that in both cases the model may not be unique. It depends on the primary choice of a class of formulas ($F(x)$ in our notation) which is chosen for the representation of the data, as well as on the approximation criterion. With this in mind, we come to the following formulation of the problem investigated in this chapter.

We are given data which are not precise (e.g., the data contain errors of measurement, or comprise numeric or linguistic expert estimations, etc.). Our goal is to construct an approximating model of the given data represented by a formula from a certain class and such that a certain criterion is satisfied.

Though we have supposed above that the given data are discrete and comprise a collection of pairs (x_i, y_i), $i = 1, \dots, n$, we will deal in fact, with a continuous function that is to be approximated. This is a usual trick in the theory of approximation which makes possible theoretical estimation of a quality of approximation. On the respective place of this chapter, we will point out, how our construction may be reduced to the discrete case.

Moreover, having an approximating model at our disposal, we would like to apply it in further investigations which may be connected with the original data. In this chapter we will demonstrate how this technique works in solving differential equations where some parameters are replaced by their approximating models.

The structure of the chapter is the following. In Sect. 3, concepts of fuzzy partition and uniform fuzzy partition of the universe are introduced. In Sects. 4 and 5, techniques of direct and inverse Fuzzy–transforms (F–transforms hereafter) are introduced and approximating properties of the inverse F–transform are established. Sect. 6 presents the technique of approximate solvability of ordinary differential equations. Two generalized methods of the Runge-Kutta class are presented. In the last subsection of Sect. 6 we showed that a numerical solution to the Cauchy problem based on F–transforms has an advantage over the classical numerical solution in a special case of noisy right-hand side.

3 Fuzzy Partition of the Universe

We take an interval $[a, b]$ as a universe. That is, all (real-valued) functions considered in this chapter have this interval as a common domain. Let us introduce fuzzy sets (given by their membership functions) which are subsets of the universe $[a, b]$ and which form a *fuzzy partition* of the universe.

Definition 1. *Let $x_1 < \ldots < x_n$ be fixed nodes within $[a, b]$, such that $x_1 = a$, $x_n = b$ and $n \geq 2$. We say that fuzzy sets $A_1, \ldots, A_n$ identified with their membership functions $A_1(x), \ldots, A_n(x)$ defined on $[a, b]$, form a* fuzzy partition *of $[a, b]$ if they fulfill the following conditions for $k = 1, \ldots, n$.*

1. *$A_k : [a, b] \to [0, 1]$, $A_k(x_k) = 1$.*
2. *$A_k(x) = 0$ if $x \notin (x_{k-1}, x_{k+1})$ where $x_{-1} = a, x_{n+1} = b$.*
3. *$A_k(x)$ is continuous.*
4. *$A_k(x)$ monotonically increases on $[x_{k-1}, x_k]$ and monotonically decreases on $[x_k, x_{k+1}]$.*
5. *$\sum_{k=1}^{n} A_k(x) = 1$, for all x.*

The membership functions $A_1(x), \ldots, A_n(x)$ are called basic functions*.*

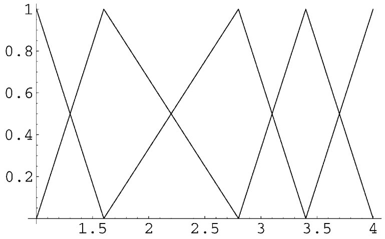


Fig. 1. An example of a fuzzy partition of $[1, 4]$ by triangular membership functions

In Fig. 1, a fuzzy partition of an interval by fuzzy sets with triangular shaped membership functions is shown. The following formulas give the formal representation of such triangular membership functions,

$$A_1(x) = \begin{cases} 1 - \frac{(x - x_1)}{h_1}, & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_k(x) = \begin{cases} \frac{(x - x_{k-1})}{h_{k-1}}, & x \in [x_{k-1}, x_k], \\ 1 - \frac{(x - x_k)}{h_k}, & x \in [x_k, x_{k+1}], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_n(x) = \begin{cases} \frac{(x - x_{n-1})}{h_{n-1}}, & x \in [x_{n-1}, x_n], \\ 0, & \text{otherwise}, \end{cases}$$

where $k = 2, \ldots n - 1$, and $h_k = x_{k+1} - x_k$.

Moreover, we say that a fuzzy partition is *uniform* if the nodes $x_1, \ldots, x_n$ are equidistant, i.e. $x_k = a + h(k - 1)$ where $h = (b - a)/(n - 1)$, $n \geq 2$ and $k = 1, \ldots, n$, and two additional properties are met,

6. $A_k(x_k - x) = A_k(x_k + x)$, for all x, $k = 2, \dots, n-1$, $n > 2$.
7. $A_{k+1}(x) = A_k(x - h)$, for all x, $k = 2, \dots, n-2$, $n > 2$.

In the case of a uniform partition, h is a length of the support of A_1 or A_n while $2h$ is the length of support of other basic functions A_k, $k = 2, \dots, n-1$. Fig. 2 shows a uniform partition by sinusoidally shaped basic functions. Their formal expressions are given below.

$$A_1(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_1) + 1), & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_k(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_k) + 1), & x \in [x_{k-1}, x_{k+1}], \\ 0, & \text{otherwise}, \end{cases}$$

where $k = 2, \dots n-1$, and

$$A_n(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_n) + 1), & x \in [x_{n-1}, x_n], \\ 0, & \text{otherwise}. \end{cases}$$

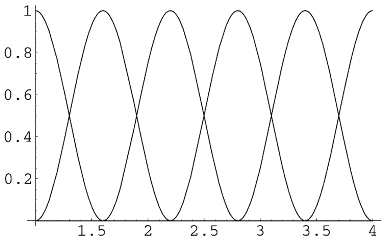


Fig. 2. An example of a uniform fuzzy partition of $[1, 4]$ by sinusoidal membership functions

For the sake of simplicity, here we consider only uniform partitions, though some results remain true in the general case. We will point out when this occurs.

The following lemma shows that, in the case of a uniform partition, the definite integral of a basic function does not depend on its concrete shape. This property will be further used to simplify a direct F–transform.

Lemma 1. *Let the uniform partition of $[a, b]$ be given by basic functions $A_1(x), \dots, A_n(x)$. Then*

$$\int_{x_1}^{x_2} A_1(x)dx = \int_{x_{n-1}}^{x_n} A_n(x)dx = \frac{h}{2}, \tag{1}$$

and for $k = 2, \dots, n-1$

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)dx = h \tag{2}$$

where h is the length of the support of A_1.

Proof. Obviously,

$$\int_{x_1}^{x_3} A_2(x)dx = \cdots = \int_{x_{n-2}}^{x_n} A_{n-1}(x)dx.$$

Therefore, to prove (2) it is sufficient to estimate

$$\int_{-h}^{h} A(x)dx,$$

where $A(x) = A_2(x + a + h)$ and $x \in [-h, h]$. Based on properties 5 and 7 of basic functions we can deduce that

$$1 - A(x) = A(x + h), x \in [-h, 0].$$

Then

$$\int_{0}^{h} A(x)dx = \int_{-h}^{0} A(x + h)dx = h - \int_{-h}^{0} A(x)dx$$

which implies (2). Equation (1) follows immediately from the symmetry of basic functions (property 6).

4 F–Transform

In this section we introduce the technique of two F–transforms: direct and inverse. The direct F–transform takes the original function (which should be at least integrable) and converts it into an n–dimensional vector. The inverse F–transform converts the n–dimensional vector into specially represented function which approximates the original one. The advantage of the direct F–transform is that it produces a simple and unique representation of an original function which enables us to use the former instead of the latter in complex computations. After finishing the computations, the result can be brought back to the space of ordinary functions by the inverse F–transform. To be sure that this can be done we need to prove a number of theorems.

The following definition (see also [6]) introduces the direct F–transform (or fuzzy transform) of a given function.

Definition 2. *Let $f(x)$ be any continuous (real-valued) function on $[a, b]$ and $A_1(x), \ldots, A_n(x)$ be basic functions which form a fuzzy partition of $[a, b]$. We say that the n–tuple of real numbers $[F_1, \ldots, F_n]$ is the F–transform of f w.r.t. $A_1, \ldots, A_n$ if*

$$F_k = \frac{\int_a^b f(x)A_k(x)dx}{\int_a^b A_k(x)dx}. \tag{3}$$

Suppose that basic functions $A_1, \ldots, A_n$ are fixed. Denote the F–transform of f w.r.t. $A_1, \ldots, A_n$ by $F_n[f]$. Then according to Definition 2, we can write,

$$F_n[f] = [F_1, \ldots, F_n]. \tag{4}$$

The elements $F_1, \ldots, F_n$ are called *components of the F-transform.*

If the partition of $[a, b]$ by $A_1, \ldots, A_n$ is uniform then the expression (2) for components of the F–transform may be simplified on the basis of Lemma 1,

$$\begin{aligned} F_1 =& \frac{2}{h} \int_{x_1}^{x_2} f(x) A_1(x) dx, \\ F_n =& \frac{2}{h} \int_{x_{n-1}}^{x_n} f(x) A_n(x) dx, \\ F_k =& \frac{1}{h} \int_{x_{k-1}}^{x_{k+1}} f(x) A_k(x) dx, \quad k = 2, \ldots, n-1. \end{aligned}$$

In the discrete case, when $f(x)$ is known only at some nodes $x_1, \ldots, x_l \in [a, b]$, the direct F–transform of f may be introduced as follows.

Definition 3. *Let $f(x)$ be given at nodes $x_1, \ldots, x_l \in [a, b]$ and $A_1(x), \ldots, A_n(x)$, $n \ll l$, be basic functions which form a fuzzy partition of $[a, b]$. We say that the n-tuple of real numbers $[F_1, \ldots, F_n]$ is the (direct) F–transform of f w.r.t. $A_1, \ldots, A_n$ if*

$$F_k = \frac{\sum_{j=1}^{l} f(x_j) A_k(x_j)}{\sum_{j=1}^{l} A_k(x_j)}.$$

It is easy to see that if a fuzzy partition (and therefore, basic functions) is fixed then the F–transform being a mapping from $C[a, b]$ (the set of all continuous functions on $[a, b]$) to $\mathbf{R}^n$ is linear, so that

$$F_n[\alpha f + \beta g] = \alpha F_n[f] + \beta F_n[g]$$

for $\alpha, \beta \in \mathbf{R}$ and functions $f, g \in C[a, b]$.

One may say that we loose information using an F–transform instead of the original function. Let us investigate this problem having on mind the following question: how fully is the original function f represented by its F–transform? First of all, we will try to estimate each component F_k, $k = 1, \ldots, n$ using different assumptions about the smoothness of f.

Lemma 2. *Let $f(x)$ be any continuous function on $[a, b]$ and $A_1(x), \ldots, A_n(x)$ be basic functions which form a uniform fuzzy partition of $[a, b]$. Then for each $k = 2, \ldots, n-1$ there exist two constants $c_{k1} \in [x_{k-1}, x_k]$ and $c_{k2} \in [x_k, x_{k+1}]$ such that*

$$F_k = \frac{1}{h} \int_{c_{k1}}^{c_{k2}} f(x) dx,$$

and for $k = 1$ ($k = n$) there exists $c \in [x_1, x_2]$ ($c \in [x_{n-1}, x_n]$) such that

$$F_1 = \frac{2}{h} \int_{x_1}^{c} f(x) dx \quad (F_n = \frac{2}{h} \int_{c}^{x_n} f(x) dx).$$

Proof. The proof can be easily obtained from the second mean–value theorem.

Therefore, by Lemma 2, we can say that F_k is a mean value of f within the interval $[c_{k1}, c_{k2}]$ and thus it accumulates the information about the function f within this interval. We can evaluate F_k more precisely if the function f is twice continuously differentiable.

Lemma 3. *Let the conditions of Lemma 2 be fulfilled, but function f be twice continuously differentiable in (a, b). Then for each $k = 1, \dots, n$*

$$F_k = f(x_k) + O(h^2). \tag{5}$$

Proof. The proof will be given for one fixed value of k which lies between 2 and $n-1$. The other two cases $k = 1$ and $k = n$ are considered analogously. We will apply the trapezoid formula with nodes x_{k-1}, x_k, x_{k+1} to the computation of the integral

$$\frac{1}{h}\int_{x_{k-1}}^{x_{k+1}} f(x)A_k(x)dx,$$

and we obtain

$$F_k = \frac{1}{h}\int_{x_{k-1}}^{x_{k+1}} f(x)A_k(x)dx =$$
$$\frac{1}{h}\cdot\frac{h}{2}\left(f(x_{k-1})A_k(x_{k-1}) + 2f(x_k)A_k(x_k) + f(x_{k+1})A_k(x_{k+1})\right) + O(h^2) =$$
$$f(x_k) + O(h^2).$$

Using (5) and applying again the trapezoid formula with nodes $x_1, \dots, x_n$ to the computation of the integral

$$\int_{x_1}^{x_k} f(x)dx,$$

we easily come to the following corollary.

Corollary 1 (Computation of Definite Integral). *Let the conditions of Lemma 3 be fulfilled and the F–transform of f be given by (4). Then for each $k = 2, \dots, n-1$,*

$$\int_{x_1}^{x_k} f(x)dx = h(\frac{1}{2}F_1 + F_2 + \cdots + F_{k-1} + \frac{1}{2}F_k) + O(h^2). \tag{6}$$

Moreover, for any continuous function $f(x)$, the integral $\int_a^b f(x)dx$ can be computed precisely,

$$\int_a^b f(x)dx = h(\frac{1}{2}F_1 + F_2 + \cdots + F_{n-1} + \frac{1}{2}F_n). \tag{7}$$

Returning to the problem of losing information by dealing with an F–transform instead of its original, we can state that an F–transform preserves mean values of the function f over $2h$ long sub–intervals with the accuracy up to h^2.

5 Inverse F–Transform

A reasonable question is the following: can we reconstruct the function by its F–transform? The answer is clear: in general not precisely, because we are losing information when changing to the direct F–transform. However, the function which can be reconstructed (by the inverse F–transform) approximates the original one in such a way that a universal convergence can be established. Moreover, the inverse F–transform fulfills the best approximation criterion that can be called the piece–wise integral least square criterion.

Definition 4. *Let $F_n[f] = [F_1, \dots, F_n]$ be the F–transform of a function $f(x)$ w.r.t. $A_1, \dots, A_n$. The function,*

$$f_{F,n}(x) = \sum_{k=1}^{n} F_k A_k(x), \tag{8}$$

will be called the inversion formula or the inverse F-transform.

The lemma below shows that the sequence of functions $f_{F,n}$ uniformly converges to f.

Lemma 4. *Let $f(x)$ be any continuous function on $[a,b]$ and let $\{(A_1^{(n)}, \dots, A_n^{(n)})_n\}$ be a sequence of uniform fuzzy partitions of $[a,b]$, one for each n. Let $\{f_{F,n}(x)\}$ be the sequence of inverse F–transforms, each with respect to the given n-tuple $A_1^{(n)}, \dots, A_n^{(n)}$. Then for any $\varepsilon > 0$ there exists n_ε such that for each $n > n_\varepsilon$ and for all $x \in [a,b]$,*

$$|f(x) - f_{F,n}(x)| < \varepsilon. \tag{9}$$

Proof. Note that the function f is uniformly continuous on $[a,b]$, i.e. for each $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$ such that for all $x_1, x_2 \in [a,b]$ $|x_1 - x_2| < \delta$ implies $|f(x_1) - f(x_2)| < \varepsilon$. To prove our lemma we choose some $\varepsilon > 0$ and find the respective δ. Let $n_\varepsilon \geq 2$ be such that $h = \frac{b-a}{n_\varepsilon - 1} \leq \frac{\delta}{2}$. We will show that with $n \geq n_\varepsilon$ (9) holds true. Let $F_{1n}, \dots, F_{nn}$ be the components of the single F–transform of f w.r.t. basic functions $A_1^{(n)}, \dots, A_n^{(n)}$, $n \geq n_\varepsilon$. Then for all $t \in [x_k, x_{k+1}]$, $k = 1, \dots, n-1$, we can evaluate,

$$|f(t) - F_{kn}| = \left| f(t) - \frac{1}{h} \int_{x_{k-1}}^{x_{k+1}} f(x) A_k^{(n)}(x)dx \right| \leq \frac{1}{h} \int_{x_{k-1}}^{x_{k+1}} |f(t) - f(x)| A_k^{(n)}(x)dx < \varepsilon,$$

and analogously,

$$|f(t) - F_{k+1,n}| < \varepsilon.$$

Therefore,

$$\left| f(t) - \sum_{k=1}^{n} F_{kn} A_k^{(n)}(t) \right| \leq \sum_{k=1}^{n} A_k^{(n)}(t) |f(t) - F_{kn}| < \varepsilon (A_k^{(n)}(t) + A_{k+1}^{(n)}(t)) = \varepsilon.$$

Because argument t has been chosen arbitrary, this proves the required inequality.

The following corollary reformulates Lemma 4.

Corollary 2. *Let the assumptions of the lemma given above be fulfilled. Then the sequence of inverse F–transforms* $\{f_{F,n}\}$ *uniformly converges to* f.

To illustrate this fact, we choose two functions with different behaviour $\sin(1/x)$ and $\sin x$ (see Figs. 3, 4) and consider different values of n. As we see below, the greater the value of n, the closer the approximating curve approaches the original function.

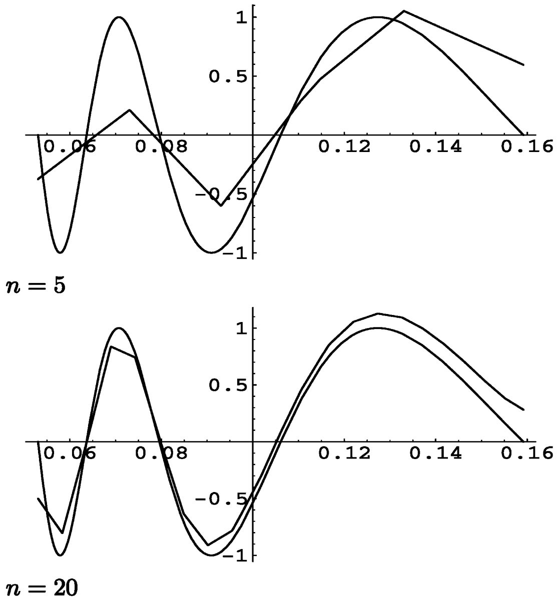


Fig. 3. $\sin(1/x)$ and its inverse transformations based on triangular shaped basic functions

Since approximation by the inverse F–transform converges uniformly to the original function, we are interested in the criterion which distinguishes it among other functions from the certain class. As discussed at the beginning, this criterion guarantees us that the inverse F–transform is the best approximation in the sense we explain below.

Let $A_1(x), \ldots, A_n(x)$ be basic functions which form a fuzzy partition of $[a, b]$, and let $f(x)$ be an integrable function on $[a, b]$. By FT we denote the class of approximating functions represented by the formula,

$$\sum_{i=1}^{n} c_i A_i(x), \tag{10}$$

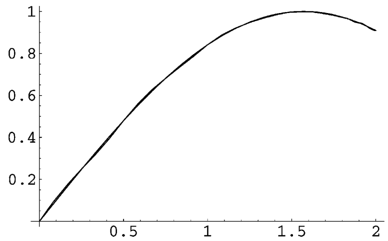


Fig. 4. sin(x) and its inverse transformation based on sinusoidally shaped basic functions ($n = 20$)

where $c_1, \ldots, c_n$ are arbitrary real coefficients. Note that our inverse F–transform belongs to FT.

Let the following piece–wise integral least square criterion

$$\Phi(c_1, \ldots, c_n) = \int_a^b \left(\sum_{i=1}^{n} (f(x) - c_i)^2 A_i(x) \right) dx, \tag{11}$$

characterizes the closeness between $f(x)$ and a function from FT. Then components $F_1, \ldots, F_n$ of the F–transform minimize (11) and therefore, determine the best approximation of $f(x)$ in FT. We leave this fact unproved because the proof is a technical exercise.

It is worth noticing that so far, we have not specified any concrete shape for basic functions. Thus, a natural question arises concerning the influence of shapes of basic functions on the quality of the approximation. We can say the following. Our criterion of the best approximation assumes that the basic functions are fixed. Therefore, properties of an approximating function are determined by the respective properties of basic functions. For example, if basic functions are of triangular shape then the approximating function will be piece–wise linear. If we choose sinusoidally shaped basic functions then the approximating function will be, of course, smoother. In any case, whatever we want to obtain from the approximating function is required from the basic functions as well. The following lemma shows how the difference between any two approximations of a given function by inverse F–transformations can be estimated. As can be seen, it depends on the character of smoothness of the original function expressed by its modulus of continuity (see below).

Lemma 5. *Let $f'_{F,n}(x)$ and $f''_{F,n}(x)$ be two inverse F–transformations of the same function $f(x)$ w.r.t. n-tuples of different basic functions, $n \geq 2$. Then $|f'_{F,n}(x) - f''_{F,n}(x)| \leq 2\omega(f, 2h)$ where ω is the modulus of continuity of $f(x)$,*

$$\omega(f, 2h) = \max_{|\delta| \leq 2h} \max_{x \in [a,b]} |f(x + \delta) - f(x)|.$$

We illustrate Lemma 5 by considering two different inverse F–transforms of functions $\sin(1/x)$ and $\sin x$. One is based on triangular-shaped basic functions and the other one is based on sinusoidal-shaped basic functions, see Figs. 5, 6. Because $\sin(1/x)$ has the modulus of continuity greater than $\sin x$, the approximation of the latter with the same value of n looks nicer (both approximations of $\sin x$ practically coincide with the original function).

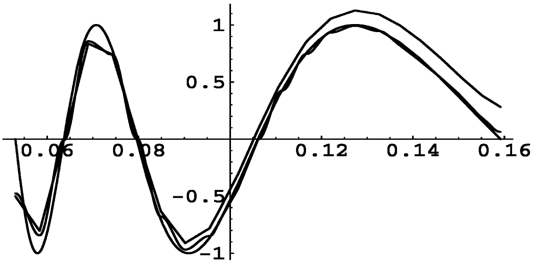


Fig. 5. $\sin(1/x)$ and two its inverse transforms based on triangular and sinusoidally shaped basic functions

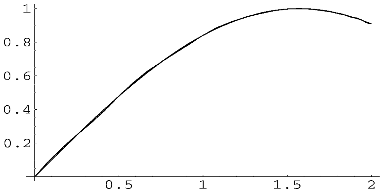


Fig. 6. $\sin x$ and two its inverse transforms based on triangular and sinusoidally shaped basic functions. Note that both approximations practically coincide with the original function

6 Approximate Solution to the Cauchy Problem

In this section we show how the approximation models based on the F–transform can be used in applications. In general, we mean by this that if the original function is replaced by an approximation model as described above, then a certain simplification of complex computations can be achieved. For demonstration, we will consider the Cauchy problem,

$$\begin{cases} y'(x) = f(x,y) \\ y(x_1) = y_1, \end{cases} \tag{12}$$

and show how it can be approximately solved in the interval $[x_1, x_n]$ if the F–transform is applied to both sides of the differential equation. Let us stress that in this section we need a uniform fuzzy partition of $[x_1, x_n]$.

6.1 The Generalized Euler Method

Suppose that we are given the Cauchy problem (12) where the ordinary functions $y(x)$ and $f(x, y(x))$ on $[x_1, x_n]$ are sufficiently smooth. Let us choose some uniform fuzzy partition of interval $[x_1, x_n]$ with parameter $h = (x_n - x_1)/(n-1)$, $n \geq 2$, and apply the direct F–transform to both parts of the differential equation. In this way we transfer the original Cauchy problem to the space of fuzzy units, solve it in the new space, and then transfer it back by the inverse F–transform. We describe the sequence of steps which leads to the solution. The justification is proved in Theorem 1.

Before we apply the direct F–transform to both parts of the differential equation, we replace $y'(x)$ by its approximation $\frac{y(x+h)-y(x)}{h}$ so that

$$y(x+h) = y(x) + hy'(x) + O(h^2). \qquad (13)$$

Denote $y_1(x) = y(x+h)$ as a new function and apply the direct F–transform to both parts of (13). By the linearity of F–transforms and Lemma 3, we obtain from (13) the expression for F–transform components of the respective functions,

$$F_n[y'] = \frac{1}{h}(F_n[y_1] - F_n[y]) + O(h^2). \qquad (14)$$

Here $F_n[y'] = [Y'_1, \dots, Y'_{n-1}]$, $F_n[y] = [Y_1, \dots, Y_{n-1}]$ and $F_n[y_1] = [Y1_1, \dots, Y1_{n-1}]$. Note that these vectors are one component shorter than in Definition 2 because the function $y_1(x)$ may not be defined on $[x_{n-1}, x_n]$. It is not difficult to prove that

$$\begin{aligned} Y1_1 &= Y_2 + O(h^2), \\ Y1_{n-1} &= Y_n + O(h^2), \\ Y1_k &= Y_{k+1}, k = 2, \dots, n-2. \end{aligned}$$

Indeed, for values $k = 2, \dots, n-2$,

$$Y1_k = \frac{1}{h}\int_{x_{k-1}}^{x_{k+1}} y(x+h)A_k(x)dx = \frac{1}{h}\int_{x_k}^{x_{k+2}} y(t)A_{k+1}(t)dt = Y_{k+1}.$$

For the values $k = 1, n-1$, the proof is analogous. Therefore, equation (14) gives us the way to compute components of the F–transform of y' via components of the F–transform of y,

$$Y'_k = \frac{1}{h}(Y_{k+1} - Y_k) + O(h^2), k = 1, \dots, n-1. \qquad (15)$$

Let us introduce $(n-1) \times n$ matrix

$$D = \frac{1}{h}\begin{pmatrix} -1 & 1 & 0 \cdots & 0 & 0 \\ 0 & -1 & 1 \cdots & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & 0 \cdots & -1 & 1 \end{pmatrix} \qquad (16)$$

so that equality (15) can be rewritten (up to $O(h^2)$) as a matrix equality,

$$F_n[y'] = DF_n[y], \tag{17}$$

where $F_n[y'] = [Y_1', \dots, Y_{n-1}']^T$, and $F_n[y] = [Y_1, \dots, Y_n]^T$.

Coming back to the Cauchy problem (12) and applying the F–transform to both sides of differential equation, we will obtain the following system of linear equations with respect to the unknown $F_n[y]$,

$$DF_n[y] = F_n[f], \tag{18}$$

where $F_n[f] = [F_1, \dots, F_{n-1}]^T$ is the F–transform of $f(x, y)$ as the function of x w.r.t. the chosen basic functions $A_1, \dots, A_n$. The last component F_n is not present in $F_n[f]$ due to the preservation of dimensionality.

Note that the system (18) does not include the initial condition of (12). For this, let us complete matrix D by adding the first row,

$$D^c = \frac{1}{h}\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

so that D^c is a $n \times n$ non–singular matrix. Analogously, let us complete the vector $F_n[f]$ by the first component $\frac{y_1}{h}$ so that

$$F_n^c[f] = \left[\frac{y_1}{h}, F_1, \dots, F_{n-1}\right]^T.$$

Then the transformed Cauchy problem can be fully represented by the following linear system of equations with respect to the unknown $F_n[y]$,

$$D^c F_n[y] = F_n^c[f]. \tag{19}$$

The solution of (19) can be given by the formula,

$$F_n[y] = (D^c)^{-1} F_n^c[f], \tag{20}$$

which in fact, is the generalized Euler method. To make sure, we compute the inverse matrix,

$$(D^c)^{-1} = h\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}$$

and rewrite (20) component–wise,

$$\begin{aligned} Y_1 &= y_1, \\ Y_2 &= y_1 + hF_1, \\ Y_3 &= y_1 + hF_1 + hF_2, \\ &\vdots \\ Y_n &= y_1 + hF_1 + \cdots + hF_{n-1}, \end{aligned}$$

or, in a more concise way,

$$\begin{aligned} Y_1 &= y_1, \\ Y_{k+1} &= Y_k + hF_k, k = 1, \dots, n-1. \end{aligned} \tag{21}$$

Formulas (21) can be applied to the computation of $Y_2, \dots, Y_n$ provided that the way of computing of $F_1, \dots, F_{n-1}$ is known. However, it cannot be done directly using formulas (3) because the expression for the function $f(x, y)$ includes also the unknown function y. Therefore, we have to go around this difficulty. The following approximation,

$$\hat{F}_k = \frac{\int_a^b f(x, Y_k) A_k(x) dx}{\int_a^b A_k(x) dx}, \tag{22}$$

for F_k, $k = 1, \dots, n-1$, is suggested. The theorem given below provides the justification.

Theorem 1. *Let the Cauchy problem (12) with twice differentiable parameters be transformed by applying the F–transform w.r.t. basic functions $A_1, \dots, A_n$ to both sides of differential equation. Then components of the F–transform of y w.r.t. the same basic functions can be approximately found from the following system of equations,*

$$Y_1 = y_1, \tag{23}$$

$$Y_{k+1} = Y_k + h\hat{F}_k, k = 1, \dots, n-1, \tag{24}$$

where $\hat{F}_k$ is given by (22). The local approximation error is of order h^2.

Proof. It has been shown that the system of linear equations (19) represents the F–transform of the Cauchy problem (12) up to $O(h^2)$. Therefore, to prove the theorem it is sufficient to show that for each $k = 1, \dots, n-1$, the order of the difference $F_k - \hat{F}_k$ is h^2. Let us denote $y(x_k) = y_k$. First, we estimate (using the trapezoid formula) the intermediate difference,

$$f(x_k, y_k) - \hat{F}_k = \frac{1}{h} \int_{x_{k-1}}^{x_{k+1}} (f(x_k, y_k) - f(x, Y_k)) A_k(x) dx =$$

$$\frac{1}{h} \cdot \frac{h}{2} \cdot 2(f(x_k, y_k) - f(x_k, Y_k)) + O(h^2) = \frac{\partial f}{\partial y}(x_k, \bar{y})(y_k - Y_k) + O(h^2),$$

where $\bar{y} \in [y_k, Y_k]$. By Lemma 3,

$$y_k - Y_k = O(h^2),$$

which, when substituted into the expression above, leads to the estimation

$$f(x_k, y_k) - \hat{F}_k = O(h^2).$$

Again by Lemma 3,

$$f(x_k, y_k) - F_k = O(h^2),$$

which, together with the previous estimation, proves that

$$F_k - \hat{F}_k = O(h^2).$$

This completes the proof.

Corollary 3. *The generalized Euler method for (12) is given by the recursive scheme (23)-(24) with local error $O(h^2)$. The approximate solution to (12) can be found by taking the inverse F–transform,*

$$y_{Y,n}(x) = \sum_{k=1}^{n} Y_k A_k(x),$$

where $A_1, \ldots, A_n$ are fixed basic functions.

Let us illustrate the generalized Euler method for the Cauchy problem (12) given by

$$\begin{cases} y'(x) = x^2 - y \\ y(x_1) = 1. \end{cases}$$

Fig. 7 shows the precise solution (gray line) and the approximate one obtained by the generalized Euler method. The global error of the approximate solution with $n = 10$ nodes has of order 10^{-1} which corresponds to the theoretical estimation.

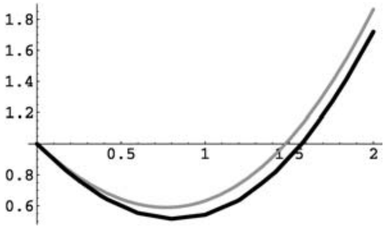


Fig. 7. Precise solution (gray line) and approximate solution (black line) of the Cauchy problem obtained by the generalized Euler method ($n = 10$)

6.2 The Generalized Euler–Cauchy Method

The generalized Euler method for the Cauchy problem has the same disadvantage as its classical prototype, namely, it is not sufficiently precise. Therefore, we will construct the generalization of more advanced method known as the Euler–Cauchy method. Recall that its classical prototype belongs to the family of the Runge–Kutta methods.

The following scheme provides us formulas for the computation of components of the F–transform of the unknown function $y(x)$ w.r.t. some basic functions $A_1, \dots, A_n$,

$$Y_1 = y_1, \tag{25}$$

$$Y^*_{k+1} = Y_k + h\hat{F}_k, \tag{26}$$

$$Y_{k+1} = Y_k + \frac{h}{2}(\hat{F}_k + \hat{F}^*_{k+1}), k = 1, \dots, n-1, \tag{27}$$

where

$$\hat{F}_k = \frac{\int_a^b f(x, \hat{Y}_k)A_k(x)dx}{\int_a^b A_k(x)dx},$$

$$\hat{F}^*_{k+1} = \frac{\int_a^b f(x, \hat{Y}^*_{k+1})A_{k+1}(x)dx}{\int_a^b A_{k+1}(x)dx}.$$

This method computes the approximate coordinates $[Y_1, \dots, Y_n]$ of the direct F–transform of the function $y(x)$. The inverse F–transform,

$$y_{Y,n}(x) = \sum_{k=1}^{n} Y_k A_k(x),$$

approximates the solution $y(x)$ of the Cauchy problem.

It can be proved that the generalized Euler–Cauchy method (25)– (27) has the local error of order h^3.

Let us illustrate the generalized Euler–Cauchy method for the Cauchy problem considered above with the same number of nodes $n = 10$. Fig. 8 shows the precise solution and the approximate one obtained by the generalized Euler–Cauchy method

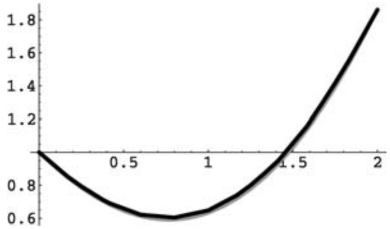


Fig. 8. Precise solution (gray line) and the approximate one (thick black line) obtained by the generalized Euler–Cauchy method. Both lines practically coincide

Remark 1. The demonstrated generalized Runge–Kutta methods can be applied to the Cauchy problem where $y(x)$ and $f(x, y)$ are vector functions, i.e., to the system of ordinary differential equations with initial values.

Remark 2. The demonstrated generalized Runge–Kutta methods can be applied to the Cauchy problem even if the function $f(x, y)$ is only partially given (at finite number of nodes) or by the description using the fuzzy "IF–THEN" rules. Formalization and presentation in the form of inverse F–transform of a description of this kind, has been discussed in [2, 3, 5].

6.3 Numerical Solution to Cauchy Problem with Noise

In this subsection, we demonstrate that a numerical solution to the Cauchy problem based on F–transforms has a significant advantages over the classical numerical solution. It concerns the case when the Cauchy problem has a noisy right-hand side,

$$\begin{cases} y'(x) = f(x, y) + \sin(\omega x + \varphi), \\ y(x_1) = y_1. \end{cases} \tag{28}$$

In this situation, classical numeric methods (we have chosen the Euler method) are unstable. Indeed, compare Fig. 9 with Fig. 10. The number of nodes increased in the second figure but the stability is still not reached. Hence, an increase in the number of nodes does not affect the stability of solution.

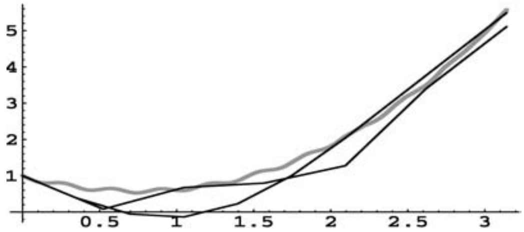


Fig. 9. Precise solution (gray line) to (28) and two numeric ones (thin black lines), both using the Euler method with the number of nodes $n = 7$ and $n = 10$ respectively. There is a visible difference between two numeric solutions

On the other hand, numeric methods based on F–transforms are stable. For illustration, we have chosen the generalized Euler method to the problem (28) given above. Fig. 11 shows the precise solution and the approximate one.

Comparison of classical numeric and F–transform methods with the same number of nodes is demonstrated below in Fig. 12. It shows the precise solution and the two approximate ones based on Euler and the generalized Euler methods with the number of nodes $n = 7$.

It is clearly seen from this example, that the approximate solution based on F–transforms gives not only a good approximation, but also eliminates a noise of the given data.

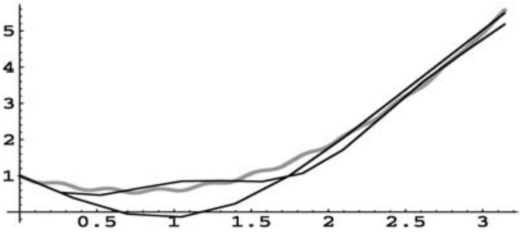


Fig. 10. Precise solution (gray line) to (28) and two numeric ones (thin black lines), both using Euler method with the number of nodes $n = 10$ and $n = 13$ respectively. There is no stability with an increase in the number of nodes

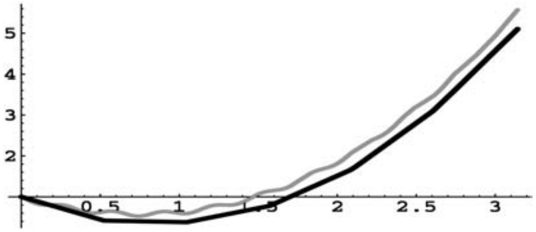


Fig. 11. Precise solution (gray line) to (28) and the approximate one obtained by the generalized Euler method (thick black line) with the number of nodes $n = 7$

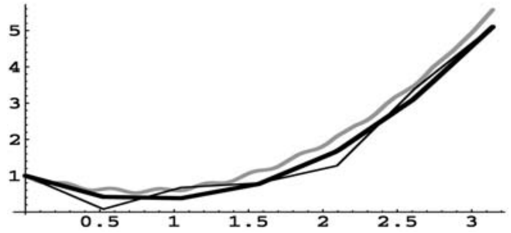


Fig. 12. Precise solution (gray line) to (28) and the two approximate ones obtained by the Euler method (thin black line) and the generalized Euler method (thick black line), both with the number of nodes $n = 7$

7 Conclusion

We have introduced a new technique of direct and inverse fuzzy transforms (F–transforms for short) which enables us to construct various approximating models depending on the choice of basic functions. The best approximation property of the inverse F–transform is established within the respective approximating space. We have demonstrated, how approximation models based on F–transforms can replace original complex functions in some computations, e.g. in integration, and in solving of differential equations. Numerical methods based on fuzzy approximation models are developed. Among them are numeric integration and approximate solution of ordinary differential equations (the Cauchy problem). Two generalized methods of the Runge–Kutta class are introduced. The advantage of the new technique based on F–transforms over some classical numeric methods is demonstrated on examples.

References

1. Novák, V., Perfilieva, I. and J. Močkoř. *Mathematical Principles of Fuzzy Logic.* Kluwer, 1999.
2. I. Perfilieva. Logical approximation: General approach to the construction of approximating formulas, in: *Proceedings EUSFLAT'2001*, Leicester, Great Britain, 2001, pp. 9–11.
3. I. Perfilieva. Neural nets and normal forms from fuzzy logic point of view. *Neural Network World* **11**, 2001, pp. 627–638.
4. I. Perfilieva Normal forms for fuzzy logic functions and their approximation ability. *Fuzzy Sets and Systems* **124**, 2001, pp. 371–384.
5. I. Perfilieva. Logical approximation. *Soft Computing* **7**, 2003 (to appear).
6. Perfilieva I., E. Chaldeeva. Fuzzy transformation, in: *Proceedings IFSA'2001 World Congress*, Vancouver, Canada, 2001.

Possible Equivalence Relations and Their Application to Hypothesis Generation in Non-deterministic Information Systems

Hiroshi Sakai

Department of Computer Engineering
Kyushu Institute of Technology
Tobata, Kitakyushu 804, Japan
sakai@comp.kyutech.ac.jp

Abstract. Non–deterministic Information Systems (*NISs*) are advanced extensions of Deterministic Information Systems (*DISs*), and *NISs* are known well as systems handling information incompleteness in tables. This paper examines manipulations on equivalence relations in *DISs* and manipulations on possible equivalence relations in *NISs*. This paper also follows rough sets based rule generation in *DISs*, and proposes rough sets based hypothesis generation in *NISs*. A hypothesis in *NISs* is defined by a formula satisfying some kinds of constraint, and effective algorithms for generating hypotheses in *NISs* are presented. Possible equivalence relations play important roles in generating hypotheses. Some illustrative examples and real executions of algorithms are shown, too.

1 Introduction

Manipulations of equivalence relations in *DISs* and manipulations of possible equivalence relations in *NISs* are examined. Such manipulations are applied to generating hypotheses in *NISs*, and a framework of hypothesis generation in *NISs* is proposed.

Rough sets theory is seen as a mathematical foundation of soft computing. The theory covers important areas of research in AI such as knowledge, imprecision, vagueness, learning and induction [1,2,3,4]. This theory usually handles tables with deterministic information, which we call *Deterministic Information Systems*. Many applications of this theory to rule generation, machine learning and knowledge discovery have been presented [3,5,6,7,8,9, 10], and several tools have been developed [3,11].

For handling incomplete information like null values [12], unknown values, missing values [13] and etc. in *DISs*, tables with non–deterministic information have been proposed [14,15,16,17], too. They are called *Non-deterministic Information Systems* or *Incomplete Information Systems*, and rough sets theory handling *DISs* has been extended to rough sets theory handling *NISs* [14,15,16,17,18]. In this extension, the concept of modality is introduced into *NISs*, and an axiomatization of logic in *NISs* has been studied.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 82–106, 2004.

However, most of work related to $NISs$ is theoretical research, especially research in logic. In $NISs$, very few work deals with algorithms toward real application. For example in [16,17], Lipski showed a question–answering system besides an axiomatization of logic. In [13], Grzymala–Busse surveyed the unknown attribute values, and studied learning from examples with unknown attribute values. In [19,20], Kryszkiewicz discussed rules in incomplete information systems. These are the most important works discussing algorithms in $NISs$.

For rough sets based data processing in $NISs$, new methods and algorithms are necessary, and especially effective manipulations of equivalence relations are needed. Rough sets based data processing in $NISs$ could also be a mathematical foundation of knowledge discovery and data mining from incomplete information. In such a situation, we are now investigating algorithms for $NISs$ including hypothesis generation [21,22,23,24,25,26].

In this paper, at first manipulations of equivalence relations in $DISs$ and some basic concepts including rule generation are studied. Then, manipulations of possible equivalence relations in $NISs$ and a framework for generating hypotheses in $NISs$ are presented. Because every NIS has information incompleteness in itself and it is impossible to know the true information, we use a term 'hypothesis generation' instead of 'rule generation'. Rough sets based useful algorithms are also examined.

2 Manipulations of Equivalence Relations in Deterministic Information Systems

Definitions concerning $DISs$, a data structure of equivalence relations and useful algorithms are shown.

2.1 Definitions Concerning DISs

A *Deterministic Information System* (DIS) is a quadruple $(OB, AT, \{VAL_a | a \in AT\}, f)$, where OB is a finite set whose elements are called *objects*, AT is a finite set whose elements are called *attributes*, VAL_a is a finite set whose elements are called *attribute values* and f is such a mapping that $f : OB \times AT \rightarrow \cup_{a \in AT} VAL_a$ which is called a *classification function*. If $f(x, a)=f(y, a)$ for any $a \in ATR \subset AT$, we see there is a relation between x and y for ATR. This relation is an equivalence relation over OB. Let $eq(ATR)$ denote this equivalence relation, and let $[x]_{ATR}$ denote an equivalence class $\{y \in OB : f(y, a)=f(x, a)$ for any $a \in ATR\}$. In this case either $[x]_{ATR}=[y]_{ATR}$ or $[x]_{ATR} \cap [y]_{ATR}=\emptyset$ holds for any $x, y \in OB$. Furthermore, $\cup_{x \in OB}[x]_{ATR}=OB$ also holds. If a set $X \subset OB$ is the union of some equivalence classes, we say X is *definable* (relative to ATR) in DIS. Otherwise we say X is *rough* (relative to ATR).

Example 1. Let us consider DIS_1 in Table 1. In DIS_1, $eq(\{color\})=\{\{x_1, x_3\}, \{x_2\}, \{x_4\}\}$ and $eq(\{size, weight\})=\{\{x_1\}, \{x_2\}, \{x_3, x_4\}\}$ hold. A set $\{x_1, x_2, x_3\}$ is definable for $\{color\}$, because $\{x_1, x_2, x_3\}=\{x_1, x_3\} \cup \{x_2\}$ holds. However, this set is rough for $\{size, weight\}$.

Table 1. A deterministic information system DIS_1.

OB	$Color$	$Size$	$Weight$
x_1	red	big	$heavy$
x_2	$blue$	$small$	$heavy$
x_3	red	$small$	$light$
x_4	$green$	$small$	$light$

2.2 A Data Structure of Equivalence Relations

For dealing with equivalence relations, we introduce a data structure. In this structure, we identify the number i with an object x_i from now on.

Two arrays $head[i]$ and $succ[i]$ $(1 \leq i \leq |OB|)$, which are correctly $head$ $[x_i]$ and $succ[x_i]$, are employed. By the order of row, it is possible to introduce the total order into OB. $head[j]$ for $j \in [i]_{ATR}$ is the first element of the equivalence class $[i]_{ATR}$, and $succ[j]$ is the successor to $j \in [i]_{ATR}$. For the last element $j \in [i]_{ATR}$, $succ[j]$=0. Every equivalence class $[i]_{ATR}$ can easily be obtained. For every i, $[i]_{ATR}$=$\{head[i], succ[head[i]], \cdots, succ[\cdots succ[head[i]]\cdots]\}$ holds. In Example 1, $eq(\{color\})$=$\{\{1,3\},\{2\},\{4\}\}$ holds. In this case, $head$[1]=1, $succ$[1]=3, $head$[2]=2, $succ$[2]=0, $head$[3]=1, $succ$[3]=0, $head$[4]=4 and $succ$[4] =0. As for object 3, $head$[3]=1, $succ$[1]=3 and $succ$[3]=0. The value 0 is the terminal symbol, and it is known that $[3]_{\{color\}} = \{1,3\}$.

Algorithm 1.
Input: A DIS with a classification function f and a set $ATR \subset AT$.
Output: Two arrays $head[i]$ and $succ[i]$ $(1 \leq i \leq |OB|)$.

```
begin
    for i:=1 to |OB| do begin head[i]:=i; succ[i]:=0 end;
    for i:=1 to (|OB|-1) do
       if head[i]=i then
          begin pre:=i;
                for j:=i+1 to |OB| do
                   if ($f(x_j,a)$=$f(x_i,a)$ for $\forall a \in ATR$) then
                   begin head[j]:=i; succ[pre]:=j; pre:=j end
          end
end.
```

To obtain an equivalence relation for any ATR in a DIS is to pick up two arrays $head[i]$ and $succ[i]$ $(1 \leq i \leq |OB|)$. By comparing with any distinct two objects, two arrays can be obtained. Algorithm 1 produces two arrays $head[i]$ and $succ[i]$.

For each pair of distinct objects $x_i, x_j \in OB$, suppose $f(x_j,a) \neq f(x_i,a)$ for some $a \in ATR$. This is the most time–consuming case, and it is necessary to compare two objects $((|OB|-1)+(|OB|-2)+\cdots+1)$ times. Therefore in the worst case, the computational complexity of Algorithm 1 is $o(|OB|^2)$. On the other hand, suppose $f(x_j,a)$=$f(x_i,a)$ for any $a \in ATR$ and any distinct objects

$x_i, x_j \in OB$. In this case, it is enough to compare two objects $(|OB|-1)$ times. This is the best case, and the complexity of Algorithm 1 is $o(|OB|)$.

2.3 An Algorithm for Merging Two Equivalence Relations

This section clarifies an algorithm to obtain $eq(A \cup B)$ from two equivalence relations $eq(A)$ and $eq(B)$.

Proposition 1 [1]. Let $A, B \subset AT$ hold in a *DIS*. The equivalence relation $eq(A \cup B)$ is $\{M \subset OB : M = [i]_A \cap [i]_B$ for $[i]_A \in eq(A)$ and $[i]_B \in eq(B)$ $(1 \leq i \leq |OB|)\}$.

Proposition 1 shows us a way to merge two equivalence relations.

Algorithm 2.
Input: Two arrays for attribute sets A and B respectively, i.e.,
$head_A[i]$, $succ_A[i]$, $head_B[i]$ and $succ_B[i]$ $(1 \leq i \leq |OB|)$.
Output: Two arrays for $A \cup B$, $head_{A\cup B}[i]$ and $succ_{A\cup B}[i]$ $(1 \leq i \leq |OB|)$.

```
begin
    for i:=1 to |OB| do
       begin head$_{A\cup B}$[i]:=i; succ$_{A\cup B}$[i]:=0 end;
    for i:=1 to |OB| do
       if head$_{A\cup B}$[i]=i then
          begin pre:=i; point:=succ$_A$[i];
                while point$\neq$ 0 do
                   begin
                      if head$_B$[point]=head$_B$[i] then
                         begin succ$_{A\cup B}$[pre]:=point;
                               head$_{A\cup B}$[point]:=i; pre:=point
                         end;
                      point:=succ$_A$[point]
                   end
          end
end.
```

If $eq(A) \subset eq(B)$ holds, $head_B[point]=head_B[i]$ always holds in the *while* loop of Algorithm 2. In this case, every object in OB is checked only once. Thus, computational complexity in the best case is $o(|OB|)$. On the contrary, if $eq(A)=\{\{1, 2, \cdots, |OB|\}\}$ and $eq(B)=\{\{1\}, \{2\}, \cdots, \{|OB|\}\}$ hold, then it is necessary to check $head_B[point]=head_B[i]$ for every *point* and i. Thus, computational complexity in the worst case is $o(|OB|^2)$.

Algorithm 2 is more effective for merging several kinds of attributes. Suppose there exist sets of attributes $A_1, A_2, \cdots, A_m$, and it is necessary to obtain $eq(A_1 \cup \cdots \cup A_m)$. In this case, Algorithm 2 is sequentially applied to $eq(A_1 \cup \cdots \cup A_k)$ and A_{k+1} $(1 \leq k \leq m-1)$. Even though the order may be $o(|OB|^2)$ in the first application of Algorithm 2, the order becomes $o(|OB|)$ in the last application of Algorithm 2, because every equivalence class $[i]_{A_1\cup\cdots\cup A_k}$ converges to a set $\{i\}$ by adding attributes.

2.4 Two Ways to Obtain Equivalence Relations

For any DIS, there are two ways to obtain an equivalence relation for any attributes. The first way is to apply Algorithm 1 directly. The second way is to apply both Algorithm 1 and 2. Namely, Algorithm 1 is applied to producing $eq(\{a_i\})$ for every $a_i \in AT$, then Algorithm 2 is sequentially applied to $eq(\cup_{1 \leq i \leq k}\{a_i\})$ and $eq(\{a_{k+1}\})$ for producing $eq(\cup_{1 \leq i \leq k+1}\{a_i\})$.

Through several experiments, the second way seems better than the first way. Especially in case of handling many kinds of ATR, the second way is more effective. For example, in case of $|OB|$=10000, $|AT|$=10, $|VAL_a|$=10 for any $a \in AT$, the first way took 0.450(sec) for producing $eq(\{a_1, a_2\})$. The second way took 0.440(sec) for producing every $eq(\{a_i\})$ and 0.080(sec) for merging. If we handle n kinds of ATR ($|ATR|$=2), the first way will take about $n \times 0.450$(sec). The second way will take about $0.440 + n \times 0.080$(sec). Therefore if $n \geq 2$, the second way is more effective than the first way. This tendency has been observed in most of $DISs$.

3 Data Dependencies in Deterministic Information Systems

Data dependencies in $DISs$ are surveyed, and the role of equivalence relations is clarified.

3.1 Definitions Concerning Data Dependencies

Let us consider two sets, $CON \subset AT$ which we call *condition attributes*, and $DEC \subset AT$ which we call *decision attributes*. Two distinct objects $x, y \in OB$ are *consistent* (*between* CON *and* DEC), if $f(x,a)$=$f(y,a)$ for every $a \in CON$ implies $f(x,a)$=$f(y,a)$ for every $a \in DEC$. A DIS is *consistent* (*between* CON *and* DEC), if every pair of distinct objects is consistent in DIS. In this case, we may say DEC *depends totally on* CON.

Now, let us show an important proposition connecting dependencies with equivalence relations. Let eq_1 and eq_2 be two equivalence relations over OB. A formula $eq_1 \subset eq_2$ means that for every equivalence class $L \in eq_1$ there exists such an equivalence class $M \in eq_2$ that $L \subset M$.

Proposition 2 [1]. For any DIS, (1) and (2) in the following are equivalent.

(1) DEC depends totally on CON.
(2) $eq(CON) \subset eq(DEC)$.

If DEC does not depend on CON, $POS_{CON}(DEC) = \cup\{L \in eq(CON)$: there exists such $M \in eq(DEC)$ that $L \subset M\}$ is applied to characterize the dependency. The degree is measured by a ratio $|POS_{CON}(DEC)|/|OB|$, and this ratio is called *the degree of dependency from* CON *to* DEC. The degree of dependency is 1 if and only if DEC depends totally on CON.

In Table 1, $POS_{\{color\}}(\{weight\})=\{2,4\} \neq OB$ and $POS_{\{size\}}(\{weight\})$ $=\{1\} \neq OB$ hold, so $\{weight\}$ does not depend on $\{color\}$ nor $\{size\}$. The degree of dependency from $\{color\}$ to $\{weight\}$ is 0.5(=2/4). On the other hand, since $eq(\{color, size\})=\{\{1\},\{2\},\{3\},\{4\}\}$ and $POS_{\{color,size\}}(\{weight\})= OB$ hold, so $\{weight\}$ depends totally on $\{color, size\}$.

3.2 An Algorithm for the Degree of Dependencies

Let $head_{CON}[i]$ and $succ_{CON}[i]$ $(1 \leq i \leq |OB|)$ be two arrays for CON, and let $head_{DEC}[i]$ and $succ_{DEC}[i]$ $(1 \leq i \leq |OB|)$ be two arrays for DEC. By using these arrays, it is easy to check $[i]_{CON} \subset [i]_{DEC}$ or not. Namely, if $head_{DEC}[j]=head_{DEC}[i]$ holds, two objects j and i must be in the same equivalence class for DEC. Therefore, if $head_{DEC}[j]=head_{DEC}[i]$ holds for any $j \in [i]_{CON}$, $[i]_{CON} \subset [i]_{DEC}$ can be concluded.

Example 2. Let $[3]_{CON}=\{1,3,5\}$ and $[3]_{DEC}=\{1,2,3,4,5\}$ hold. Here, $1(= head_{CON}[3]) \in [3]_{CON}$ and $head_{DEC}[1]=head_{DEC}[3]$ hold, so $1 \in [3]_{DEC}$. For object $3 \in [3]_{CON}$, $3 \in [3]_{DEC}$ is trivial. For object $5(=succ_{CON}[3]) \in [3]_{CON}$, $head_{DEC}[5]=head_{DEC}[3]$ holds, and $5 \in [3]_{DEC}$ can also be concluded. Because $succ_{CON}[5]=0$, it is known that $[3]_{CON} \subset [3]_{DEC}$.

Algorithm 3.
Input: Any object x_i and two arrays for CON and DEC, respectively.
Output: $[i]_{CON} \subset [i]_{DEC}$ or not.

```
begin
   mark:=0; point:=head$_{CON}$[i];
   while point$\neq$0 do
      if head$_{DEC}$[point]=head$_{DEC}$[i] then
         point:=succ$_{CON}$[point] else begin mark:=1; point:=0 end;
   if mark=0 then [i]$_{CON}$ $\subset$[i]$_{DEC}$ else [i]$_{CON}$ $\not\subset$[i]$_{DEC}$
end.
```

Algorithm 4.
Input: Two arrays for CON and DEC, respectively.
Output: The degree of dependency from CON to DEC.

```
begin
   count:=0;
   for i:=1 to |OB| do
      if head$_{CON}$[i]=i then
         if [i]$_{CON}$ $\subset$[i]$_{DEC}$ then count:=count+|[i]$_{CON}$|;
   degree:=count/|OB|
end.
```

In Algorithm 3, every element $point \in [i]_{CON}$ is sequentially picked up, and $head_{DEC}[point]=head_{DEC}[i]$ is checked. Thus, computational complexity in the worst case is $o(|[i]_{CON}|)$. In Algorithm 4, the condition $head_{CON}[i]=i$ implies object i is the first element in the class $[i]_{CON}$. For such an object i, the condition $head_{DEC}[point]=head_{DEC}[i]$ is checked $|[i]_{CON}|$ times. Namely in the worst case, this condition is checked $\sum_{i:head_{CON}[i]=i}|[i]_{CON}|$=$|OB|$ times. Thus, complexity in the worst case of Algorithm 4 is $o(|OB|)$.

4 Rule Generation in Deterministic Information Systems

Rule generation in *DISs* is briefly surveyed.

4.1 Previous Work

Let us survey works on generating rules in *DISs*. For any *DIS*, let *CON* be condition attributes, and let *DEC* be decision attributes. For any object $x_i \in OB$, $imp(x_i, CON, DEC)$ denotes a formula called *implication*:

$\wedge_{a \in CON}[a, f(x_i, a)] \Rightarrow \wedge_{a \in DEC}[a, f(x_i, a)]$,

where $[a, f(x_i, a)]$ implies that $f(x_i, a)$ is the value of the attribute a. This is called a *descriptor* in [16]. In most of works, a *rule* in a *DIS* is defined by an implication satisfying some kinds of constraint.

The most familiar constraint is the dependency from *CON* to *DEC*. If the degree of dependency is more than a threshold value, each implication in a *DIS* is seen as a rule [1,2]. VPRS by Ziarko [8] and LERS by Grzymala–Busse [7] enhanced the constraint of the dependency from *CON* to *DEC*. Some criteria are also applied to each implication as other constraint. In [9], three criteria in the following are applied to any ψ=$imp(x_i, CON, DEC)$:

$support(\psi)=|\{x_j \in OB | imp(x_j, {\scriptstyle CON,DEC})=\psi\}|/|OB|$,
$accuracy(\psi)=|[i]_{CON} \cap [i]_{DEC}|/|[i]_{CON}|$,
$coverage(\psi)=|[i]_{CON} \cap [i]_{DEC}|/|[i]_{DEC}|$.

If each criterion value of an implication is more than each threshold value, this implication is seen as a rule.

In Table 1, $eq(\{color, size\})=\{\{1\}, \{2\}, \{3\}, \{4\}\}$ and $eq(\{weight\})=\{\{1, 2\}, \{3, 4\}\}$ hold. Since $eq(\{color, size\}) \subset eq(\{weight\})$ holds, $\{weight\}$ depends totally on $\{color, size\}$, and rules like

$[color, red] \wedge [size, big] \Rightarrow [weight, heavy]$

are generated. Then for $CON=\{color\}$ and $DEC=\{weight\}$, $POS_{\{color\}}(\{weight\})=\{2, 4\} \neq OB$, and $\{weight\}$ does not depend on $\{color\}$. However, an implication $imp(x_4, \{color\}, \{weight\})$ is seen as a rule under such a constraint that *support* $\geq$0.25, *accuracy*=1.0 and *coverage* $\geq$ 0.5. These three values are usually given by users, and we do not discuss how to decide these values.

4.2 An Algorithm for Criterion Values of Implications

Algorithm 5 shows a way to calculate three criteria values, *support*, *accuracy* and *coverage*. Two arrays $head[i]$ and $succ[i]$ take important roles in Algorithm 5.

Algorithm 5.
Input: Object x_i and two arrays for CON and DEC, respectively.
Output: *support*, *accuracy* and *coverage* values of $imp(x_i, CON, DEC)$.

```
begin
    numer:=0; denom:=0; point:=head$_{CON}$[i];
    while point$\neq$0
       begin denom:=denom+1;
             if head$_{DEC}$[point]=head$_{DEC}$[i] numer:=numer+1;
             point:=succ$_{CON}$[point]
       end;
    support:=numer/|OB|; accuracy:=numer/denom;
    numer:=0; denom:=0; point:=head$_{DEC}$[i];
    while point$\neq$0
       begin denom:=denom+1;
             if head$_{CON}$[point]=head$_{CON}$[i] numer:=numer+1;
             point:=succ$_{DEC}$[point]
       end;
    coverage:=numer/denom
end.
```

For *support* and *accuracy*, $head_{DEC}[point]$=$head_{DEC}[i]$ is checked $|[i]_{CON}|$ times, and $head_{CON}[point]$=$head_{CON}[i]$ is checked $|[i]_{DEC}|$ times in *coverage*. Thus, the order of Algorithm 5 is $o(|[i]_{CON}| + |[i]_{DEC}|)$. If we do not use equivalence relations for this calculation, two distinct objects must be compared. For every value, $f(x_i, a)$=$f(x_j, a)$ ($a \in CON \cup DEC$, $i \neq j$) is checked. The computational order is $o(2 \times |OB|)$ for any $x_i \in OB$. If it is possible to use equivalence relations, Algorithm 5 is much better than the way without using equivalence relations.

5 Manipulations of Possible Equivalence Relations in Non-deterministic Information Systems

Definitions on $NISs$, possible equivalence relations and useful algorithms are shown. A procedure to obtain all possible equivalence relations for any attributes in any $NISs$ is also presented.

5.1 Definitions Concerning NISs and Possible Equivalence Relations

A *Non-deterministic Information System* (NIS) is also a quadruple (OB, $AT, \{VAL_a | a \in AT\}, g$), where g is such a mapping that $g : OB \times AT \rightarrow P(\cup_{a \in AT} VAL_a)$ (a power set of $\cup_{a \in AT} VAL_a$). Every set $g(x, a)$ is interpreted as that there is an actual value in this set but it is not known. This is the unknown interpretation for the incomplete information. Especially if the actual value is not known at all, $g(x, a)$ is equal to VAL_a. This is the same as null value interpretation.

Definition 1. *Suppose there exist a NIS=$(OB, AT, \{VAL_a | a \in AT\}, g)$, a set $ATR \subset AT$ and such a mapping $h : OB \times ATR \rightarrow \cup_{a \in ATR} VAL_a$ that $h(x, a) \in g(x, a)$. We call such a DIS=$(OB, ATR, \{VAL_a | a \in ATR\}, h)$ a derived DIS (for ATR) from NIS. We also call every equivalence relation in a derived DIS a possible equivalence relation (pe-relation), and we call every element in a pe-relation a possible equivalence class (pe-class).*

Example 3. Let us consider NIS_1 in Table 2. In NIS_1, there are $36(=2^2 \times 3^2)$ derived $DISs$ for ATR=$\{A, B, C\}$. Table 3 shows a derived DIS from NIS_1. In this DIS, tuples are pairwise different and $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$ is the equivalence relation. Namely, $\{\{ 1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$ is a *pe*-relation in NIS_1. The sets $\{1\}, \cdots, \{8\}$ are *pe*-classes in NIS_1. There are 36 derived $DISs$ for NIS_1, so there may exist 36 different *pe*-relations. However, in NIS_1, the above *pe*-relation is unique. Table 4 shows a derived DIS for ATR=$\{A, B\}$. In this DIS, the equivalence is $\{\{1\}, \{2, 7\}, \{3, 4, 6\}, \{5\}, \{8\}\}$. There are $18(=2 \times 3^2)$ derived $DISs$, and there are four kinds of *pe*-relations.

Table 2. A non-deterministic information system NIS_1.

OB	A	B	C
x_1	3	1	1
x_2	5	$\{2, 4\}$	2
x_3	$\{1, 4, 5\}$	5	4
x_4	4	5	2
x_5	3	5	2
x_6	4	5	$\{1, 3\}$
x_7	5	4	1
x_8	1	$\{1, 3, 4\}$	1

Table 3. A derived deterministic information system from NIS_1.

OB	A	B	C
x_1	3	1	1
x_2	5	4	2
x_3	1	5	4
x_4	4	5	2
x_5	3	5	2
x_6	4	5	1
x_7	5	4	1
x_8	1	1	1

Table 4. A derived deterministic information system for $\{A, B\}$ from NIS_1.

OB	A	B
x_1	3	1
x_2	5	4
x_3	4	5
x_4	4	5
x_5	3	5
x_6	4	5
x_7	5	4
x_8	1	1

5.2 Definability of a Set in $NISs$ and Possible Equivalence Relations

The definability of a set in $DISs$ is extended to the definability of a set in $NISs$.

Definition 2. *For any NIS and any set $X \subset OB$, X is definable in NIS if X is definable in some derived $DISs$ from NIS.*

To check this, a simple way is to examine the definability of a set for all derived $DISs$. However, this way will not be suitable for $NISs$ with large number of derived $DISs$. We propose another method according to the next proposition.

Proposition 3 [26]. Suppose there exist a NIS and a set $X \subset OB$. X is definable in NIS, if and only if there exist subsets $CL_1, \cdots, CL_m$ of OB, satisfying (1) and (2) below:

(1) $\cup_i CL_i = X$,
(2) $\{CL_1, \cdots, CL_m\}$ is a subset of a *pe*-relation.

Algorithm 6 solves the definability of a set according to Proposition 3.

Algorithm 6.

```
Input: A $NIS$ and a set $X \subset OB$.
Output: $X$ is definable in this $NIS$ or not.
(1) $X^*=X$ and $eq=\emptyset$.
(2) For any element $x \in X^*$, find a set $CL$ satisfying two conditions below:
      (CL-1) $x \in CL$ and $CL \subset X^*$,
      (CL-2) $eq \cup \{CL\}$ is a subset of a pe-relation.
(2-1) If there is a set $CL$, then $[x]=CL$, $eq=eq \cup \{[x]\}$ and $X^*=X^* - CL$.
      If $X^* \neq \emptyset$ then go to (2). If $X^*=\emptyset$ then $X$ is definable.
(2-2) If there is no $CL$, then backtrack. If there is no branch for backtracking,
      then $X$ is not definable in this $NIS$.
```

This algorithm is similar to LEM1 and LEM2 algorithms [27]. The set CL is unique in every DIS, but in NIS this set CL need not be unique. Therefore, a search with backtracking is applied. At this point, Algorithm 6 is different from LEM1 and LEM2.

Algorithm 6 has the following merit. After solving the definability of a set, a subset of a *pe*-relation is stored in the variable *eq* as a side effect. If $X=OB$, it is possible to obtain a *pe*-relation by this variable *eq*. By applying this algorithm repeatedly until there is no branch for backtracking, all *pe*-relations can be obtained.

5.3 Some Properties of Possible Equivalence Classes

This section clarifies some properties of *pe*-classes according to [26]. Algorithm 6 has been realized by using these properties.

Definition 3. *Let us consider a NIS=$(OB, AT, \{VAL_a : a \in AT\}, g)$ and a set ATR=$\{a_1, \cdots, a_n\} \subset AT$. For any $x \in OB$, $PT(x, ATR)$ denotes the Cartesian product $g(x, a_1) \times \cdots \times g(x, a_n)$. We call every element a possible tuple (for ATR) of x.*

Every possible tuple ζ=$(\eta_{a_1}, \cdots, \eta_{a_n}) \in PT(x, ATR)$ is identified with a formula $\bigwedge_{1 \leq i \leq n}[a_i, \eta_{a_i}]$.

Definition 4. *Let us consider a NIS=$(OB, AT, \{VAL_a : a \in AT\}, g)$ and a set $ATR \subset AT$. For every $x \in OB$ and every possible tuple ζ of x, two sets inf and sup are defined as follows:*

(1) $inf(x, \zeta, ATR)$=$\{x\} \cup \{y \in OB : PT(y, ATR)$=$\{\zeta\}\}$.
(2) $sup(x, \zeta, ATR)$=$\{y \in OB : \zeta \in PT(y, ATR)\}$.

A set *sup* is equal to the set defined by the similarity relation SIM in [15,20]. In our framework, both *sup* and inf are necessary for characterizing *pe*-classes.

Theorem 1. *Let us consider a NIS=$(OB, AT, \{VAL_a : a \in AT\}, g)$ and a set $ATR \subset AT$. For any object $x \in OB$, conditions (1) and (2) in the following are equivalent.*

(1) A set X such that $x \in X$ is a pe-class for ATR.
(2) $inf(x, \zeta, ATR) \subset X \subset sup(x, \zeta, ATR)$ for some $\zeta \in PT(x, ATR)$.

Theorem 1 shows us a way to pick up the set CL in Algorithm 6. In Algorithm 6, *pe*-classes $CL_1, \cdots, CL_m$ are sequentially fixed, and every CL_i must satisfy condition (2) in Theorem 1.

Furthermore, it is necessary to remark that the set CL_m depends upon previous sets $CL_1, \cdots, CL_{m-1}$. Now, we deal with this problem.

Definition 5. *Suppose condition (2) in Theorem 1 holds. In this case, we say every object in X is positively applied to $\zeta \in PT(x, ATR)$ and every object in $sup(x, \zeta, ATR) - X$ is negatively applied to $\zeta \in PT(x, ATR)$.*

Definition 6. *Two lists $PLIST$ and $NLIST$ are recursively defined as follows:*

Initial step: $PLIST$=$NLIST$=$\emptyset$.

Recursive step: Condition (2) in Theorem 1 is applied to finding a pe-class CL in Algorithm 6.

In this application, two lists are revised as follows:

$PLIST$=$PLIST \cup \{[x, \zeta] | x$ is positively applied to $\zeta\}$,
$NLIST$=$NLIST \cup \{[x, \zeta] | x$ is negatively applied to $\zeta\}$.

Either positive or negative application of every object is stored in $PLIST$ and $NLIST$, and these two lists control the search with backtracking in Algorithm 6.

Definition 7. *Let us consider a NIS, a set $ATR \subset AT$ and a sequence of pe-classes. If $PLIST$ and $NLIST$ by this sequence satisfy the next three conditions, we say this sequence is proper.*

(1) $PLIST \cap NLIST = \emptyset$.
(2) For any $x \in OB$, $\{[x,\zeta] : \zeta \in PT(x, ATR)\} \cap PLIST$ is either an empty set or a singleton set.
(3) For any $x \in OB$, $\{[x,\zeta] : \zeta \in PT(x, ATR)\} - NLIST \neq \emptyset$.

Theorem 2. *Let us consider a NIS, $ATR \subset AT$ and a sequence of pe-classes $CL_1, \cdots, CL_m$. The (1) and (2) in the following are equivalent.*
(1) This sequence is proper.
(2) There is such a pe-relation eq that $\{CL_1, \cdots, CL_m\} \subset eq$.

In order to realize (CL-2) in Algorithm 6, Theorem 2 is applied. Namely, for a candidate CL in (CL-1) and a proper sequence $CL_1, \cdots, CL_m$, the sequence $CL_1, \cdots, CL_m, CL$ is checked, proper or not. If so, a new proper sequence $CL_1, \cdots, CL_m, CL_{m+1}(=CL)$ is obtained. Otherwise, this candidate CL is rejected.

5.4 A Simulation of Algorithm 6

This subsection clarifies roles of Theorem 1 and Theorem 2.

Table 5. A non-deterministic information system NIS_2.

OB	A
x_1	1
x_2	2
x_3	$\{1, 2\}$

In Table 5, there are clearly two *pe*-relations $\{\{1,3\},\{2\}\}$ and $\{\{1\},\{2,3\}\}$, and, $inf(1,(1),\{A\})=\{1\}$, $sup(1,(1),\{A\})=\{1,3\}$, $inf(2,(2),\{A\})=\{2\}$, $sup(2,(2),\{A\})=\{2,3\}$, $inf(3,(1),\{A\})=sup(3,(1),\{A\})=\{1,3\}$, $inf(3,(2),\{A\})=sup(3,(2),\{A\})=\{2,3\}$. For element $1 \in X=\{1,2,3\}$, there are two *pe*-classes $\{1\}$ and $\{1,3\}$ by Theorem 1. Two *pe*-relations are obtained as follows.

(CASE 1) For a selection of $\{1\}$, $X=\{2,3\}$, $PLIST=\{[1,(1)]\}$, $NLIST=\{[3,(1)]\}$ and $eq=\{\{1\}\}$ are derived. For the first element $2 \in X=\{2,3\}$, there are two *pe*-classes $\{2\}$ and $\{2,3\}$ by Theorem 1. However, the *pe*-class $\{2\}$ can not be applicable, because in this case $NLIST=\{[3,(1)],[3,(2)]\}$ contradicts condition (3) in Definition 7. The *pe*-class $\{2,3\}$ is only applicable to $X=\{2,3\}$, and $X=\emptyset$, $PLIST=\{[1,(1)],[2,(2)],[3,(2)]\}$, $NLIST=\{[3,(1)]\}$ and $eq=\{\{1\},\{2,3\}\}$ are derived.

(CASE 2) For a selection of $\{1,3\}$, $X=\{2\}$, $PLIST=\{[1,(1)],[3,(1)]\}$, $eq=\{\{1,3\}\}$ and $NLIST=\emptyset$ are derived. For the first element $2 \in X=\{2\}$, there are two *pe*-classes $\{2\}$ and $\{2,3\}$ by Theorem 1. However, *pe*-class $\{2,3\}$ can not be applicable, because in this case $PLIST=\{[1,(1)],[2,(2)],[3,(1)],[3,(2)]\}$ contradicts condition (2) in Definition 7. The *pe*-class $\{2\}$ is only applicable to $X=\{2\}$, and $X=\emptyset$, $PLIST=\{[1,(1)],[2,(2)],[3,(1)]\}$, $NLIST=\{[3,(2)]\}$ and $eq=\{\{1,3\},\{2\}\}$ are derived.

5.5 A Real Execution to Obtain All Possible Equivalence Relations for Any Attributes

A real execution by realized programs is shown. Every step is implemented on a workstation with 450MHz UltraSparc CPU in prolog and C. Because a search with backtracking is necessary in Algorithm 6, prolog is employed. The total procedure consists of four steps.

Procedure 1. (A procedure to obtain all *pe*-relations for any attributes)
(Step 1) Prepare a data file.
(Step 2) Translate data to an internal expression, like inf and sup, for each attribute.
(Step 3) Obtain all *pe*-relations for each attribute by using Algorithm 6.
(Step 4) Fix a set of attributes ATR, and produce all *pe*-relations by using Algorithm 2.

The following is the real execution for ATR=$\{A, B\}$ in Table 2. It is enough to do Step 2 and 3 only once, and it is enough to do Step 4 for each ATR. The contents of the data file for NIS_1 in Table 2 are as follows:

```
object(8,3), data(1,[3,1,1]), data(2,[5,[2,4],2]),
data(3,[[1,4,5],5,4]), data(4,[4,5,2]), data(5,[3,5,2]),
data(6,[4,5,[1,3]]), data(7,[5,4,1]), data(8,[1,[1,3,4],1]).
```

Realized programs can deal with any data file in this syntax.

```
?-translate.
File Name:'data.pl'.
EXEC_TIME = 0.045(sec)
yes
```

Program *translate* in Step 2 outputs three files, A.rs, B.rs and C.rs. Every file stores the internal expression for each attribute.

```
?-pe.
File Name:'data.pl'.
EXEC_TIME = 0.055 (sec)
yes
```

Program *pe* in Step 3 outputs three files, A.pe, B.pe and C.pe. Every file stores all *pe*-relations. To execute program *merge* in Step 4, it is necessary to prepare a file *merge.dat* whose contents are the name of output file, the number of attributes and the names of files to merge. The following is the real execution of merging A.pe and B.pe.

```
% merge
Merging A.pe ...
Merging B.pe ...
EXEC_TIME = 0.000(sec)
%
```

After executing *merge*, the following four *pe*-relations for attributes $\{A, B\}$ are obtained.

```
[1] [[1],[2],[3,4,6],[5],[7],[8]] 3
[2] [[1],[2],[3],[4,6],[5],[7],[8]] 6
[3] [[1],[2,7],[3,4,6],[5],[8]] 3
[4] [[1],[2,7],[3],[4,6],[5],[8]] 6
```

It is also known that the first *pe*-relation $\{\{1\},\{2\},\{3,4,6\},\{5\},\{7\},\{8\}\}$ occurs in three derived $DISs$ of 18 derived $DISs$. In this way, it is possible to obtain all *pe*-relations for any ATR in any NIS.

6 Hypothesis Generation in Non-deterministic Information Systems

This section advances the concept of rules from $DISs$ to $NISs$.

6.1 A Definition of Hypotheses in NISs

Definition 8. *For any NIS, let CON be condition attributes and let DEC be decision attributes. For any $x \in OB$, $PI(x, CON, DEC)$ denotes a set $\{\zeta \Rightarrow \eta : \zeta \in PT(x, CON),\ \eta \in PT(x, DEC)\}$. We call an element $\zeta \Rightarrow \eta$ a possible implication from x.*

Let us consider a case that $CON{=}\{A, B\}$ and $DEC{=}\{C\}$ in Table 2. $PT(x_2, \{A, B\}){=}\{(5, 2), (5, 4)\}$, $PT(x_2, \{C\}){=}\{(2)\}$ and $PI(x_2, \{A, B\}, \{C\})$ consists of two possible implications $[A, 5] \wedge [B, 2] \Rightarrow [C, 2]$ and $[A, 5] \wedge [B, 4] \Rightarrow [C, 2]$.

Definition 9. *For any NIS, let CON be condition attributes and let DEC be decision attributes. A hypothesis from an object x is a possible implication ψ such that $\psi \in PI(x, CON, DEC)$ and ψ satisfies some kinds of constraint. Especially, if ψ is a hypothesis and $PI(x, CON, DEC){=}\{\psi\}$, we call this ψ a rule from object x.*

Constraints for possible implications characterize hypotheses in $NISs$. In the subsequent sections, constraints in $DISs$ are extended to constraints in $NISs$, and they are applied to possible implications.

6.2 Constraint 1: Data Dependencies in NISs

Dependencies in $DISs$ are extended to dependencies in $NISs$. This dependency is applied to CON and DEC as a constraint.

Definition 10. *Let us consider a NIS, condition attributes CON, decision attributes DEC and all derived $DISs$ for $CON \cup DEC$. Let NUM be the number of all derived $DISs$. For two threshold values val_1 and $val_2 (0 \leq val_1 \leq 1,\ 0 \leq val_2 \leq 1)$, if conditions (Dep 1) and (Dep 2) hold then DEC depends on CON in NIS.*

(Dep 1) $min_k\{$degree of dependency in $DIS_k (1 \leq k \leq NUM)\} \geq val_1$.

(Dep 2) $|\{DIS_k | DIS_k (1 \leq k \leq NUM)$ is consistent$\}|/NUM \geq val_2$.

In Definition 10, condition (Dep 1) specifies the minimum value of the degree of dependency, and condition (Dep 2) specifies the ratio of consistent $DISs$ to NUM. For such a value that $val_2 \geq 0.9$, more than 90% of derived $DISs$ must be consistent. We see these are conditions of dependency from CON to DEC.

6.3 Constraint 2: Six Groups of Possible Implications in NISs

Six groups of possible implications are introduced in $NISs$. These groups are also useful for characterizing each possible implication.

Definition 11. *Let us consider a NIS, condition attributes CON and decision attributes DEC. For any $\psi \in PI(x, CON, DEC)$ $(x \in OB)$, $DD(\psi, x, CON, DEC)$ denotes a set $\{\phi : \phi$ is such a derived DIS for $CON \cup DEC$ that an implication $imp(x, CON, DEC)$ in ϕ is equal to $\psi\}$.*

Definition 12. *Let us consider a NIS, condition attributes CON and decision attributes DEC. If $PI(x, CON, DEC)$ $(x \in OB)$ is a singleton set $\{\psi\}$, we say ψ is $definite$. Otherwise we say each $\psi \in PI(x, CON, DEC)$ is $indefinite$. If a set $\{\phi \in DD(\psi, x, CON, DEC) |$ x is consistent with other objects in $\phi\}$ is equal to $DD(\psi, x, CON, DEC)$, we say ψ is globally consistent. If a set $\{\phi \in DD(\psi, x, CON, DEC) |$ x is consistent with other objects in $\phi\}$ is equal to $\emptyset$, we say ψ is globally inconsistent. Otherwise we say ψ is $marginal$. Finally, six groups in Table 6 are defined.*

Table 6. Six groups of possible implications in $NISs$.

	$Globally\_consistent$	$Marginal$	$Globally\_inconsistent$
$Definite$	DGC	DM	DGI
$Indefinite$	IGC	IM	IGI

Example 4. In Table 2, there exist $18(= 2 \times 3^2)$ derived $DISs$ for attributes $\{A, B\}$, and $PI(x_3, \{A\}, \{B\})$ consists of three possible implications. For the first possible implication $\psi_1 : [A, 1] \Rightarrow [B, 5]$, $DD(\psi_1, x_3, \{A\}, \{B\})$ consists of 6(=18/3) derived $DISs$. Because x_3 is inconsistent with x_8 in all 6 derived $DISs$, ψ_1 belongs to IGI group. For the second possible implication $\psi_2 : [A, 4] \Rightarrow [B, 5]$, $DD(\psi_2, x_3, \{A\}, \{B\})$ also consists of 6(=18/3) derived $DISs$. Because x_3 is consistent with other objects in all 6 derived $DISs$, ψ_2 belongs to IGC group.

There exists no information incompleteness for each possible implication in definite groups. Possible implications in DGC or IGC groups, which correspond to the *certain rules* in [20], are preferable. Possible implications in DGI nor IGI groups are inappropriate. These six groups are applied as another constraint to $NISs$.

6.4 Constraint 3: Three Criteria in NISs

Criteria *support*, *accuracy* and *coverage* for implications in $DISs$ are extended to criteria for possible implications in $NISs$. These criteria are also applied to possible implications as a constraint.

Definition 13. *Let us consider a NIS, condition attributes CON and decision attributes DEC. For any $\psi \in PI(x, CON, DEC)$ $(x \in OB)$,*

$min\_sup(\psi) = min_{\phi \in DD(\psi,x,CON,DEC)}\{support(\psi)\ in\ \phi\}$,
$max\_sup(\psi) = max_{\phi \in DD(\psi,x,CON,DEC)}\{support(\psi)\ in\ \phi\}$.
$mean\_sup(\psi) = \frac{1}{|DD(\psi,x,CON,DEC)|}\sum_{\phi \in DD(\psi,x,CON,DEC)} support(\psi)$.

Similarly $min\_acc(\psi)$, $max\_acc(\psi)$ and $mean\_acc(\psi)$ denote minimum, maximum and mean values of $accuracy(\psi)$, and $min\_cov(\psi)$, $max\_cov(\psi)$ and $mean\_cov(\psi)$ denote minimum, maximum and mean values of $coverage(\psi)$.

We see that values *max*, *min* and *mean* characterize each possible implication.

Example 5. In Table 2, there exist $36(= 2^2 \times 3^2)$ derived $DISs$ for attributes $\{A, B, C\}$, and $PI(x_2, \{A, B\}, \{C\})$ consists of two possible implications. For the second possible implication $\psi_2 : [A, 5] \wedge [B, 4] \Rightarrow [C, 2]$, $DD(\psi_2, x_2, \{A, B\}, \{C\})$ consists of 18(=36/2) derived $DISs$. Table 3 is an element of $DD(\psi_2, x_2, \{A, B\}, \{C\})$, and in this table $support(\psi_2)$=1/8, $accuracy(\psi_2)$=1/2 and $coverage(\psi_2)$= 1/3 hold. Based on *support*, *accuracy* and *coverage* values in every 18 derived DIS, three values *max*, *min* and *mean* of ψ_2 are calculated.

6.5 A Definition of Hypothesis Generation in NISs

Three concepts, i.e., the dependency, six groups of possible implications and three criteria, are applied to defining hypothesis generation in $NISs$.

Problem 1. (Hypothesis Generation in NISs)
For any NIS, any decision attributes DEC, find all $x \in OB$ and all $\psi \in PI(x, CON, DEC)$ such that

(Hypo 1) There exists a dependency from CON to DEC.
(Hypo 2) ψ belongs to DGC or IGC.
(Hypo 3) ψ satisfies conditions of criteria *support*, *accuracy* and *coverage*.

Even though there may be other definitions of hypotheses, from now on we deal with hypotheses specified in Problem 1.

7 Algorithms for Hypothesis Generation in Non-deterministic Information Systems

This section examines rough sets based algorithms for realizing a hypothesis generator in $NISs$. Namely, algorithms for conditions (Hypo 1), (Hypo 2) and (Hypo 3) in Problem 1 are discussed.

7.1 An Algorithm for Data Dependencies in NISs

In order to check the dependency from CON to DEC, conditions (Dep 1) and (Dep 2) in Definition 10 are examined. It is possible to check these conditions by calculating each degree of dependency in every derived DIS. However, this way is not suitable for $NISs$ with several derived $DISs$. We propose an algorithm by using *pe*-relations, Proposition 2 and Algorithm 4.

Definition 14. *Let $pe(CON, t)$ $(1 \leq t \leq n)$ be all kinds of pe-relations for CON, and let $penum(CON, t)$ denote the number of elements in a set $\{\phi : \phi$ is a derived DIS whose equivalence relation is equal to $pe(CON, t)\}$. $pe(DEC, s)$ and $penum(DEC, s)$ $(1 \leq s \leq m)$ are similarly defined.*

According to Procedure 1 in Section 5.5, it is possible to obtain these *pe*-relations. By using these *pe*-relations, conditions (Dep 1) and (Dep 2) in Definition 10 can easily be calculated.

Algorithm 7. Suppose conditions in Definition 10 hold.

Input: $pe(CON, t)$, $penum(CON, t)$ $(1 \leq t \leq n)$ and $pe(DEC, s)$, $penum(DEC, s)$ $(1 \leq s \leq m)$

Output: $min{=}min_k\{$degree of dependency in $DIS_k\}$ and $ratio{=}|\{DIS_k | DIS_k$ is consistent$\}|/NUM$.

```
begin
    min:=1.0; ratio:=0.0;
    for t:=1 to n do
    for s:=1 to m do
    begin
       calculate a degree by $pe(CON, t)$, $pe(DEC, s)$ and Algorithm 4;
       if degree=1.0 then ratio:=ratio+$penum(CON, t) \times penum(DEC, s)$;
       if degree < min then min:=degree
    end;
    ratio:=ratio/NUM
end.
```

Algorithm 7 repeats the same part $n \times m$ times. These numbers n and m are kinds of *pe*-relations for CON and DEC respectively, and the number of all kinds of *pe*-relations for CON and DEC is usually smaller than the number of all derived $DISs$. Algorithm 7 makes use of this property. Some preliminary programs for the dependency and the execution time for some $NISs$ have already been presented in [24,25,26].

7.2 Possible Equivalence Relations Related to Possible Tuples

Suppose conditions in Algorithm 7 hold. In this case, it is easy to realize a program *ratio*, whose execution is as follows:

```
% ratio
File Name for Condition: AB.pe
File Name for Decision: C.pe
Consistent ratio of object 1: 1.000(=36 / 36)
Consistent ratio of object 2: 0.500(=18 / 36)
          :       :       :
Consistent ratio of object 7: 0.500(=18 / 36)
Consistent ratio of object 8: 1.000(=36 / 36)
EXEC_TIME = 0.000(sec)
```

In this execution, the dependency from $\{A, B\}$ to $\{C\}$ in Table 2 is specified, and the consistent ratio of each object is responded. There are $36(=2^2 \times 3^2)$ derived $DISs$ for attributes $\{A, B, C\}$, and it is known that objects x_1 and x_8 are always consistent. Objects x_2 and x_7 are consistent in 18 derived $DISs$. It is possible to calculate such a ratio by using $pe(CON, t)$, $penum(CON, t)$ and $pe(DEC, s)$, $penum(DEC, s)$ $(1 \leq t \leq n, 1 \leq s \leq m)$.

However in this execution, the relation between possible implications and the consistent ratio is not clear. For example, $PI(x_1, \{A, B\}, \{C\})$ is a singleton set, and this possible implication belongs to DGC group. On the other hand, $PI(x_8, \{A, B\}, \{C\})$ consists of three elements. Each possible implication belongs to IGC group. Furthermore for any possible implication ψ, it is impossible to calculate three criteria like $support(\psi)$ by using such pe-relations. In order to deal with six groups of possible implications and three criteria, link information between possible implications and pe-relations is needed.

Definition 15. *Let us consider a* NIS *and a set* $ATR \subset AT$*. For any* $\zeta \in PT(x, ATR)$*, a pe-relation related to* ζ *is an equivalence relation* $eq(ATR)$ *in a derived* DIS *whose tuple of* x *is equal to* ζ*. Let* $RELpe(\zeta, x, ATR)$ *denote a set of all pe-relation related to* ζ*.*

Example 6. Let us consider the second possible implication $\psi_2 : [A, 5] \wedge [B, 4] \Rightarrow [C, 2] \in PI(x_2, \{A, B\}, \{C\})$ in Table 2. As we have already shown in Section 5.5, there exist four kinds of pe-relations $pe(\{A, B\}, t)$ $(1 \leq t \leq 4)$ for $CON=\{A, B\}$. The pe-relations related to $\zeta : [A, 5] \wedge [B, 4] \in PT(x_2, \{A, B\})$ are the following two of four pe-relations:

$pe(\{A, B\}, 3)$: `[[1],[2,7],[3,4,6],[5],[8]] 3`
$pe(\{A, B\}, 4)$: `[[1],[2,7],[3],[4,6],[5],[8]] 6`

Similarly, the pe-relations related to $\eta : [C, 2] \in PT(x_2, \{C\})$ are the following two of two pe-relations:

$pe(\{C\}, 1)$: `[[1,6,7,8],[2,4,5],[3]] 1`
$pe(\{C\}, 2)$: `[[1,7,8],[2,4,5],[3],[6]] 1`

Now we briefly refer how to obtain $RELpe(\zeta, x, ATR)$ for any $ATR \subset AT$, any $x \in OB$ and any $\zeta \in PT(x, ATR)$. This procedure relies on Algorithm 6. Two lists $PLIST$ and $NLIST$ are employed for realizing Algorithm 6. By checking the definability of a set OB, a pe–relation is stored in the variable eq. At the same time, information for a derived DIS is stored in $PLIST$, and every $[x, \zeta] \in PLIST$ specifies a possible tuple of $x \in OB$. Such information is stored in a file, and $RELpe(\zeta, x, CON)$ for each $\zeta \in PT(x, CON)$ and $RELpe(\eta, x, DEC)$ for each $\eta \in PT(x, DEC)$ are produced. We have realized this program by revising Algorithm 6.

7.3 Algorithms for Six Groups and Three Criteria

For any $\psi(= \zeta \Rightarrow \eta) \in PI(x, CON, DEC)$, two sets $RELpe(\zeta, x, CON)$ and $RELpe(\eta, x, DEC)$ are fixed. By applying adjusted Algorithm 7 to each pair of $pe(CON, t) \in RELpe(\zeta, x, CON)$ and $pe(DEC, s) \in RELpe(\eta, x, DEC)$ repeatedly, it is possible to decide one of six groups where ψ belongs to.

Similarly, by applying Algorithm 5 to each pair of $pe(CON, t) \in RELpe(\zeta, x, CON)$ and $pe(DEC, s) \in RELpe(\eta, x, DEC)$ repeatedly, it is possible to calculate three criterion values.

Example 7. Let us continue the contents in Example 6, and consider $\psi_2 : [A, 5] \wedge [B, 4] \Rightarrow [C, 2] \in PI(x_2, \{A, B\}, \{C\})$ in Table 2.

Adjusted Algorithm 7 is applied to each pair of $pe(\{A, B\}, t)$ $(3 \leq t \leq 4)$ and $pe(\{C\}, s)$ $(1 \leq s \leq 2)$. For a pair of $pe(\{A, B\}, 3)$ and $pe(\{C\}, 1)$, $[2]_{\{A,B\}}=\{2, 7\} \in pe(\{A, B\}, 3)$ and $[2]_{\{C\}}=\{2, 4, 5\} \in pe(\{C\}, 1)$ hold. In this case $[2]_{\{A,B\}} \not\subset [2]_{\{C\}}$ holds, and it is known that objects x_2 and x_7 are inconsistent. Similarly, x_2 and x_7 are inconsistent in every pair of $pe(\{A, B\}, t)$ and $pe(\{C\}, s)$. Namely, ψ_2 appears in 18(=36/2) derived *DISs* and x_2 is inconsistent in 18 derived *DISs*. Therefore, it is possible to conclude that ψ_2 belongs to *IGI* group.

Similarly for a pair of $pe(\{A, B\}, 3)$ and $pe(\{C\}, 1)$, $|[2]_{\{A,B\}} \cap [2]_{\{C\}}|=|\{2\}|=1$ holds. In this way,

$$support(\psi_2)=|[2]_{\{A,B\}} \cap [2]_{\{C\}}|/|OB|=1/|OB|=1/8,$$
$$accuracy(\psi_2)=|[2]_{\{A,B\}} \cap [2]_{\{C\}}|/|[2]_{\{A,B\}}|=1/2,$$
$$coverage(\psi_2)=|[2]_{\{A,B\}} \cap [2]_{\{C\}}|/|[2]_{\{C\}}|=1/3$$

are easily obtained. Since there are four pairs of $pe(\{A, B\}, t)$ and $pe(\{C\}, s)$, this procedure is repeated 4 times, and *min*, *max* and *mean* values are obtained.

8 A Real Execution

Let us consider NIS_3 in Table 7, which was produced by using a random number program. This section shows a hypothesis generation for specified decision attributes $\{E\}$ in Table 7.

The real execution generally consists of two phases below:

Phase 1. For fixed $DEC(=\{E\})$, find *CON* such that *DEC* depends on *CON*.

Phase 2. For fixed *CON* and *DEC*, produce link information $RELpe(\zeta, x, CON)$ and $RELpe(\eta, x, DEC)$ for each ζ, η and $x \in OB$. Then, fix criterion values of *min_sup*, *max_sup*, $\cdots$, *mean_cov*. Finally pick up possible implications, which satisfy the conditions of criterion values and belong to *DGC* or *IGC* groups.

8.1 Phase 1

In Fig. 1, program *translate* creates new files *A.rs*, $\cdots$, *E.rs*. Every file stores *internal expressions* for each attribute. Program *pe_display* creates new files *A.pe*, $\cdots$, *E.pe*. Every file stores all kinds of *pe*-relations for each attribute.

For checking the dependency from *CON* to *DEC*, all kinds of *pe*-relations for *CON* and *DEC* are produced at first. Then, program *depend* is executed. In Fig. 2 and Fig.3, program *merge* is executed for $CON=\{A, B, C, D\}$ and $CON=\{B, C\}$, respectively. Every condition for executing program *merge* is stored in a file named *merge.dat*.

Table 7. A non-deterministic information system NIS_3.

OB	A	B	C	D	E
x_1	1	1	4	2	$\{1,2,5\}$
x_2	$\{2,4,5\}$	5	3	2	4
x_3	5	2	2	$\{1,5\}$	2
x_4	$\{1,2\}$	3	4	$\{2,5\}$	$\{1,2,3\}$
x_5	$\{2,3,4\}$	$\{1,2\}$	$\{1,3,5\}$	$\{1,4\}$	5
x_6	2	1	5	$\{1,4\}$	$\{1,4,5\}$
x_7	1	$\{1,3,4\}$	1	$\{2,3,4\}$	2
x_8	$\{2,4,5\}$	4	4	$\{2,3,4\}$	$\{3,5\}$

8.2 Phase 2

Suppose a dependency from $\{B,C\}$ to $\{E\}$ be observed, and CON be fixed to $\{B,C\}$ for DEC=$\{E\}$. In Fig. 4, *pe*-relations related to possible implication are obtained. Then, criterion values are fixed as follows: *min_sup* $\geq$0.1, *mean_sup* $\geq$0.1, *min_acc* $\geq$0.5, *mean_acc* $\geq$0.6, *min_cov* $\geq$0.5, *mean_cov* $\geq$ 0.7. Fig. 5 shows an execution of program *hypotheses*. There exist three possible implications satisfying constraints. For attributes $\{B,C,E\}$, there exist 972 derived $DISs$. The first hypothesis from object x_2 appears 972 derived $DISs$, and it is consistent in all 972 derived $DISs$. Namely it is not only a hypothesis but also a rule. As for object x_4, $|PI(x_4,\{B,C\},\{E\})|$=3 holds, and only one possible implication satisfies constraints.

9 Remarks on a Tool and Execution Time

For obtaining more reliable hypotheses, the degrees of dependency and threshold values like *min_sp*, *min_ac* and *min_co* are fixed at high level. For obtaining lots of hypotheses, the degrees of dependency and threshold values are fixed at low level. By tuning such values, it is possible to generate hypotheses in any NIS.

In Phase 1, all kinds of *pe*-relations are extracted by *pe_display* command. This is a time–consuming program, but it is enough to execute this program only once. Because, each *pe*-relation for any CON is produced by *merge* command. Both the orders of *merge* and *depend* are almost $o(|OB|)$, therefore it takes less execution time in Phase 1.

Now, we show the experimental result in Phase 2. Four data files, NIS_4, NIS_5, NIS_6 and NIS_7, are produced by using a random numbers program. In every $NIS_i(4 \leq i \leq 7)$, $|AT|$=5 and $|VAL_a|$=10 for any $a \in AT$, and the sizes of OB are 30, 50, 70 and 100 respectively. About 20% of attribute values in NIS_4, 10% of attribute values in NIS_5 and 5% of attribute values in NIS_6 and NIS_7 are non–deterministic.

Table 8 shows execution time for generating hypotheses according to CON= $\{A,B\}$ and DEC=$\{E\}$. In NIS_4 there exist 4 kinds of *pe*–relations for 1944

```
?-translate.
File Name:'data2.pl'.
EXEC_TIME=0.137(sec)
yes
?-pe_display.
File Name:'data2.pl'.
Attribute A
[1] [[1,4,7],[2,3,8],[5,6]] 1
[2] [[1,4,7],[2,3,8],[5],[6]] 2
      :     :     :
Attribute B
      :     :     :
Attribute E
[1] [[1,3,4,7],[2,6],[5],[8]] 1
      :     :     :
[50] [[1,6],[2],[3,7],[4],[5,8]] 1
POSSIBLE CASES 54
EXEC_TIME=0.890(sec)
yes
```

Fig. 1. An execution for obtaining all *pe*-relations in each attribute.

```
% merge
Merging A.pe ...
Merging B.pe ...
Merging C.pe ...
Merging D.pe ...
EXEC_TIME=0.010(sec)
% depend
File for Condition:ABCD.pe
File for Decision:E.pe
CRITERION 1
Minimum Degree of Dependency:0.750
Maximum Degree of Dependency:1.000
CRITERION 2
Derived DISs:7558272
Derived Consistent DISs:7418304
Degree of Consistent DISs:0.981
EXEC_TIME=0.000(sec)
%
```

Fig. 2. An execution for the dependency $\{A, B, C, D\} \Rightarrow \{E\}$.

derived $DISs$ and 36 kinds of pe–relations for 36 derived $DISs$. There are totally 69984(=1944 × 36) derived $DISs$ for $\{A, B, E\}$, and there exist 144(=4 × 36) pairs of $pe(\{A, B\}, t)$ $(1 \leq t \leq 4)$ and $pe(\{E\}, s)$ $(1 \leq s \leq 36)$. Similarly, there exist 31104(=576 × 54), 26244(=1458 × 18) and 279936(=1296 × 216) derived

```
% merge
Merging B.pe ...
Merging C.pe ...
EXEC_TIME=0.000(sec)
% depend
File for Condition:BC.pe
File for Decision:E.pe
CRITERION 1
Minimum Degree of Dependency:0.750
Maximum Degree of Dependency:1.000
CRITERION 2
Derived DISs:972
Derived Consistent DISs:810
Degree of Consistent DISs:0.833
EXEC_TIME=0.000(sec)
%
```

Fig. 3. An execution for the dependency $\{B, C\} \Rightarrow \{E\}$.

```
?-translate_attrib.
File Name:'data2.pl'.
Attribute File:'attrib.pl'.
EXEC_TIME=0.040(sec)
yes
?-link(con).
EXEC_TIME=0.033(sec)
yes
?-link(dec).
EXEC_TIME=0.570(sec)
yes
```

Fig. 4. An execution for linking possible tuples to *pe*-relations.

$DISs$ in NIS_5, NIS_6 and NIS_7, respectively. According to Table 8, program *link* occupies most execution time. After making links between possible tuples and *pe*–relations, program *hypotheses* is executed. Since each three criterion value is fixed to 0, most of possible tuples are displayed in every execution. In NIS_7, it took much execution time for displaying hypotheses.

Table 9 shows execution time of Phase 2 and a simple method. The column of simple method shows (execution time to obtain criterion values in a DIS)×(the number of derived $DISs$). However, this value is meaningless, because it does not include the time of handling every derived DIS. The execution time of this method will be much worse. Table 9 shows the effect of using possible equivalence relations.

```
% hypotheses
Hypothesis:[B,C]⇒[E]
Hypothesis from object 2
   [5,3]⇒[4] 1.000(=972/972,972:DGC)
   MINSUP: 0.125, MAXSUP: 0.125, MEANSUP: 0.125
   MINACC: 1.000, MAXACC: 1.000, MEANACC: 1.000
   MINCOV: 0.500, MAXCOV: 1.000, MEANCOV: 0.833
Hypothesis from object 4
   [3,4]⇒[3] 1.000(=324/324,972:IGC)
   MINSUP: 0.125, MAXSUP: 0.125, MEANSUP: 0.125
   MINACC: 1.000, MAXACC: 1.000, MEANACC: 1.000
   MINCOV: 0.500, MAXCOV: 1.000, MEANCOV: 0.750
Hypothesis from object 8
   [4,4]⇒[3] 1.000(=486/486,972:IGC)
   MINSUP: 0.125, MAXSUP: 0.125, MEANSUP: 0.125
   MINACC: 1.000, MAXACC: 1.000, MEANACC: 1.000
   MINCOV: 0.500, MAXCOV: 1.000, MEANCOV: 0.833
EXEC_TIME=0.010(sec)
%
```

Fig. 5. An execution for generating hypotheses.

Table 8. Execution time(sec) for CON={A,B} and DEC={E}.

NIS	$Translate\_attrib$	$Link(con)$	$Link(dec)$	$Hypotheses$
NIS_4	0.15	9.79 (4/1944)	0.22 (36/36)	0.31 (144/69984)
NIS_5	0.22	5.12 (60/576)	0.56 (54/54)	0.77 (3240/31104)
NIS_6	0.32	17.85 (486/1458)	0.34 (18/18)	6.40 (8748/26244)
NIS_7	0.55	21.97 (216/1296)	4.40 (216/216)	18.82 (46656/279936)

10 Concluding Remarks

Equivalence relations in $DISs$ are extended to *pe*-relations in $NISs$, and manipulations on both equivalence relations and possible equivalence relations are presented. Then, a framework of hypothesis generation in $NISs$ is proposed, and realized. In this realization, possible equivalence relations take important roles.

In $NISs$, the computation of (Hypo 1), (Hypo 2) and (Hypo 3) is generally proportional to the number of all derived $DISs$, and our tool is suitable for

Table 9. Execution time(sec) by two methods.

NIS	$Phase\_2$	$Simple\_method$
NIS_4	10.47	$69.98(= 0.001 \times 69984)$
NIS_5	6.67	$93.31(= 0.003 \times 31104)$
NIS_6	24.92	$104.97(= 0.004 \times 26244)$
NIS_7	45.75	$2519.42(= 0.009 \times 279936)$

$NISs$ with large number of derived $DISs$. This comes from the fact that the number of all kinds of *pe*-relations is usually much smaller than the number of all derived $DISs$.

Up to now, there are only few works that deal with real data processing in $NISs$. For handling not only certain information but also incomplete information, the research of rough sets based data processing in $NISs$ will be important. Furthermore, rough sets based data processing in $NISs$ could also be a mathematical foundation of knowledge discovery and data mining from incomplete information. We are now planning to analyze students' life style information by questionnaires. This data consists of about 2500 objects and about 100 attributes. Most of VAL_a $(a \in AT)$ are $\{worst, bad, normal, good, best\}$, and this data contains lots of void information for some disagreeable questions.

References

1. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer, 1991.
2. Pawlak, Z. New look on Bayes' Theorem - the rough set outlook. *Bulletin of Int. Rough Set Society* **5**, 2001, pp. 1–8.
3. Komorowski, J., Pawlak, Z., Polkowski, L., and Skowron, A. Rough Sets: A tutorial, in: *Rough Fuzzy Hybridization* (S. K. Pal, A. Skowron (eds.)). Springer Singapore, 1999, pp. 3–98
4. Nakamura, A., Tsumoto, S., Tanaka, H., and Kobayashi, S. Rough set theory and its applications. *Journal of Japanese Society for AI* **11**, 1996, pp. 209–215
5. Polkowski, L., Skowron, A.(eds.). *Rough Sets in Knowledge Discovery 1.* Studies in Fuzziness and Soft Computing **18**, Physica, 1998.
6. Polkowski, L., Skowron, A.(eds.). *Rough Sets in Knowledge Discovery 2.* Studies in Fuzziness and Soft Computing **19**, Physica, 1998.
7. Grzymala–Busse, J. A new version of the rule induction system LERS. *Fundamenta Informaticae* **31**, 1997, pp. 27–39.
8. Ziarko, W. Variable precision rough set model. *Journal of Computer and System Sciences* **46**, 1993, pp. 39–59.
9. Tsumoto, S. Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic. *Information Sciences* **124**, 2000, pp. 125–137.
10. Zhong, N., Dong, J., Fujitsu, S., and Ohsuga, S. Soft techniques to rule discovery in data. *Transactions of Information Processing Society of Japan* **39**, 1998, pp. 2581–2592.
11. Rough Set Software. *Bulletin of Int. Rough Set Society* **2**, 1998, pp. 15–46.

12. Codd, E. A relational model of data for large shared data banks. *Communications of the ACM* **13**, 1970, pp. 377–387.
13. Grzymala–Busse, J. On the unknown attribute values in learning from examples, in: *Proceedings ISMIS'91. Lecture Notes in Artificial Intelligence* 542. Springer, 1991, pp. 368–377.
14. Orlowska, E., Pawlak, Z. Representation of nondeterministic information. *Theoretical Computer Science* **29**, 1984, pp. 27–39.
15. Orlowska, E. What you always wanted to know about rough sets, in: *Incomplete Information: Rough Set Analysis*(Orlowska, E. (ed.)). Studies in Fuzziness and Soft Computing **13**. Physica, 1998, pp. 1–20.
16. Lipski, W. On semantic issues connected with incomplete information Databases. *ACM Transactions on Database Systems* **4**, 1979, pp. 262–296.
17. Lipski, W. On databases with incomplete information. *Journal of the ACM* **28**, 1981, pp. 41–70.
18. Nakamura, A. A rough logic based on incomplete information and its application. *Int. Journal of Approximate Reasoning* **15**, 1996, pp. 367–378.
19. Kryszkiewicz, M. Rough set approach to incomplete information systems. *Information Sciences* **112**, 1998, pp. 39–49.
20. Kryszkiewicz, M. Rules in incomplete information systems. *Information Sciences* **113**, 1999, pp. 271–292.
21. Sakai, H. Another fuzzy prolog, in: *Proceedings of the 4th Int. Workshop on Rough Set, Fuzzy Set and Machine Discovery*, Tokyo, 1996, pp. 261–268.
22. Sakai H. Some issues on nondeterministic knowledge bases with incomplete and selective information, in: *Proceedings RSCTC'98. Lecture Notes in Artificial Intelligence* **1424**, Springer, 1998, pp. 424–431.
23. Sakai, H., Okuma, A. An algorithm for finding equivalence relations from tables with non–deterministic information, in: *Proceedings RSFDGrC'99. Lecture Notes in Artificial Intelligence* **1711**, Springer, 1999, pp. 64–72.
24. Sakai, H., Okuma, A. An algorithm for checking dependencies of attributes in a table with non–deterministic information: A rough sets based approach, in: *Proceedings PRICAI'00. Lecture Notes in Artificial Intelligence* **1886**, Springer, 2000, pp. 219–229.
25. Sakai, H. Two procedures for dependencies among attributes in a table with nondeterministic information: a summary, in: *Proceedings JSAI'01. Lecture Notes in Artificial Intelligence* **2253**, Springer, 2001, pp. 301–305.
26. Sakai, H. Effective procedures for handling possible equivalence relations in non–deterministic information systems. *Fundamenta Informaticae* **48**, 2001, pp. 343–362.
27. Grzymala–Busse, J., Werbrouck, P. On the best search method in the LEM1 and LEM2 algorithms, in: *Incomplete Information: Rough Set Analysis* (Orlowska, E. (ed.). Studies in Fuzziness and Soft Computing **13**, Physica, 2001, pp. 75–91.

Applications of Fuzzy Logic Functions to Knowledge Discovery in Databases

Noboru Takagi[1], Hiroaki Kikuchi[2], and Masao Mukaidono[3]

[1] Department of Electronics and Informatics, Toyama Prefectural University,
Kosugi-machi, Imizu-gun, Toyama 939-0398, Japan
[2] Department of Information Media Technology, Tokai University,
Hiratsuka-shi, Kanagawa 259-1292, Japan
[3] Department of Computer Science, Meiji University,
Tama-ku, Kawasaki-shi, Kanagawa 214-8571, Japan

Abstract. This chapter is a summary of knowledge discovery algorithms that take an input of training examples of target knowledge, and output a fuzzy logic formula that best fits the training examples. The execution is done in three steps; first, the given mapping is divided into some Q-equivalent classes; second, the distances between the mapping and each local fuzzy logic function are calculated by a simplified logic formula; and last, the shortest distance is obtained by a modified graph-theoretic algorithm. After a fundamental algorithm for fitting is provided, fuzzy logic functions are applied to a more practical example of classification problem, in which expressiveness of fuzzy logic functions is examined for a well-known machine learning database.

1 Introduction

Functions on the unit interval $[0, 1]$ which can consist of logical operations min, max, and $x \mapsto 1-x$ are called fuzzy logic functions (or used to be referred as fuzzy switching functions). Research on fuzzy logic functions have been extensively done by many researchers [2–7, 10, 11] since L. A. Zadeh proposed fuzzy set theory. Theoretical aspects of fuzzy logic functions were discussed especially in the 1970s and 1980s. The motivation for fuzzy logic functions in those days was centered on the possibility of their applications to switching circuits. Thus, the canonical sum-of-products forms, the minimal sum-of-product forms, and the prime-implicant expressions, etc., have been central issues of research on fuzzy logic functions as well as logic designs based on Boolean functions. This paper will, however, summarize a new application of fuzzy logic functions to knowledge discovery in databases (KDD).

A new generation of intelligent tools for automated data mining and knowledge discovery has been needed since the 1990s, because of the rapid progress of computer systems and our capabilities for collecting and storing data of all kinds. By this background, many KDD techniques have been investigated to handle collections of data samples. Some of them are based on soft computing

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 107–128, 2004.

methodologies such as neural networks, fuzzy set theory, rough set theory, and decision trees, etc., etc..

For example, by applying the back–propagation procedure, multi–layered neural networks can find a non–linear function which has little error between the function and a data set. Theoretically, if we have an appropriate number of hidden units, we can find a representation that can perform any mapping from input to output through these hidden units.

KDD is concerned with identifying interesting patterns and describing them in a concise and meaningful manner [19]. However, multi–layered neural networks are not adequate for describing interesting patterns in a concise and meaningful way, because the result of the error back–propagating procedure is given as a non–linear function based on sigmoidal functions. Fuzzy logic functions are attractive in achieving this high comprehensibility requirement for knowledge representations, because they are expressed by logic formulas, which can be considered as knowledge representations similar to our natural language.

The paper will discuss algorithms which can determine a fuzzy logic function which is approximately equal to a numerical data set. Here, any numerical data set dealt with in this paper will be assumed to be an incompletely specified function $f : A \to [0, 1]$, where A is a finite subset of $[0, 1]^n$. Because fuzzy logic functions consist of only three kinds of logical operations (i.e., min, max, and $x \mapsto 1 - x$), they are not powerful enough to express any functions on $[0, 1]$. Thus, our aim is to find a fuzzy logic function which is approximately equal to an incompletely specified function. Our first approach is to clarify an algorithm that enables us to find an optimal fuzzy logic function. This algorithm is built by the divide–and – conquer method, since the number of fuzzy logic functions asymptotically increases in $O(2^{3^n})$ with n variables. Second, fuzzy logic functions are applied to a more practical problem, that is, classification problem. To complete the problem, a genetic algorithm (GA) will be applied to find a near optimal fuzzy logic function.

The chapter is constructed as follows. In Sect. 2, definitions and mathematical foundations of fuzzy logic functions will be given. Then, Sect. 3 will discuss an algorithm for finding an optimal fuzzy logic function. In Sect. 3, some of the mathematical properties will be also discussed, because they are needed to build the algorithm. In Sect. 4, an algorithm based on GA will be shown. This algorithm will discover classification rules of a data set. The performance of the algorithm proposed will be examined by using a well–known machine learning database, and also will be compared with other existing classification methods.

2 Definitions and Basic Properties of Fuzzy Logic Functions

This section describes definitions and some mathematical properties of fuzzy logic functions.

Let V, V_2 and V_3 be the unit interval $[0, 1]$, the binary set $\{0, 1\}$ and the ternary set $\{0, 0.5, 1\}$, respectively. We then define logical operations $\wedge$, $\vee$ and $\neg$ below,

$$\begin{cases} x \wedge y = \min(x, y), \\ x \vee y = \max(x, y), \\ \neg x \quad = 1 - x, \end{cases} \tag{1}$$

where $x, y \in V$.

Definition 1 Logic formulas can be defined inductively as follows.

(1) The constants 0 and 1, and any variable x_i $(i = 1, 2, \ldots, n)$ are logic formulas.
(2) If g and h are logic formulas, then $(g \wedge h)$, $(g \vee h)$ and $(\neg g)$ are logic formulas.
(3) It is a logic formula if and only if we get it from (1) and (2) in a finite number of steps.

Definition 2 A mapping f from V^n into V is a *fuzzy logic function* if and only if there exists at least one logic formula which expresses the mapping f.

Next, we define a binary relation $\succeq$ over the set V.

Definition 3 Let a and b be elements of V. Then, the relation $a \succeq b$ holds if and only if either $0.5 \geq a \geq b$ or $b \geq a \geq 0.5$ holds. The relation $\succeq$ can be extended to V^n by letting $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ be elements of V^n, $\boldsymbol{a} \succeq \boldsymbol{b}$ if and only if $a_i \succeq b_i$ for each i $(i = 1, \ldots, n)$. Any two elements a in $[0, 0.5)$ and b in $(0.5, 1]$ are not comparable with respect to $\succeq$. We denote this by $a \not\succeq b$. We write $a \succ b$ to mean that $a \succeq b$ but $a \neq b$.

It is evident that the relation $\succeq$ is a partial order relation on V^n.

Definition 4 Let x and λ be elements of V. A *quantization* $\overline{x}^{\lambda}$ of x by λ is an element of V_3 defined by

$$\overline{x}^{\lambda} = \begin{cases} 0 & \text{if } 0 \leq x \leq \min(\lambda, 1 - \lambda) \text{ and } x \neq 0.5, \\ 1 & \text{if } 1 \geq x \geq \max(\lambda, 1 - \lambda) \text{ and } x \neq 0.5, \\ 0.5 & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be an element of V^n. A quantization $\overline{\boldsymbol{x}}^{\lambda}$ of $\boldsymbol{x}$ by λ is defined by $\overline{\boldsymbol{x}}^{\lambda} = (\overline{x_1}^{\lambda}, \ldots, \overline{x_n}^{\lambda}) \in V_3^n$.

Definition 5 Any variable x_i and its negation $\neg x_i$ are called *literals.* A product (i.e., $\wedge$-ing) of some literals will be called a *simple product term* (or *simple term* for short), if there exists at most one literal for each i. A *complementary product term* (or *complementary term* for short) is a product of some literals that contains both positive and negative literals for some x_i. A sum ($\vee$-ing) of some products is called *sum-of-products form*, if no product term appears more than twice in it.

It is proved that any fuzzy logic function can be expressed by a sum-of-products form [4]. Further, in writing product terms, the operation $\wedge$ is often omitted.

Definition 6 Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ be elements of V_3^n and $V_3^n - V_2^n$, respectively. A simple term α^a corresponding to $\boldsymbol{a}$ and a complementary term β^b corresponding to $\boldsymbol{b}$ are defined by $\alpha^a = x_1^{a_1} \wedge \cdots \wedge x_n^{a_n}$ and $\beta^b = x_1^{b_1} \wedge \cdots \wedge x_n^{b_n}$, where

$$x_i^{a_i} = \begin{cases} x_i & \text{if } a_i = 1 \\ \neg x_i & \text{if } a_i = 0 \\ 1 & \text{if } a_i = 0.5, \end{cases} \qquad x_i^{b_i} = \begin{cases} x_i & \text{if } b_i = 1 \\ \neg x_i & \text{if } b_i = 0 \\ x_i \neg x_i & \text{if } b_i = 0.5, \end{cases}$$

for every i $(i = 1, \ldots, n)$.

Theorem 1 (Normality) [11] Let f be a fuzzy logic function. Then, $\boldsymbol{a} \in V_2^n$ implies $f(\boldsymbol{a}) \in V_2$.

Theorem 2 (Monotonicity) [11] Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be elements of V^n. Then, $\boldsymbol{a} \succeq \boldsymbol{b}$ implies $f(\boldsymbol{a}) \succeq f(\boldsymbol{b})$.

Theorem 3 (Quantization) [11] Let f be a fuzzy logic function, and $\boldsymbol{a}$ be an element of V^n. Then, $\overline{f(\boldsymbol{a})}^\lambda = f(\overline{\boldsymbol{a}}^\lambda)$ holds for all λ in V.

Theorem 4 [11] Let f and g be fuzzy logic functions. Then, $f(\boldsymbol{a}) = g(\boldsymbol{a})$ for every element $\boldsymbol{a}$ of V_3^n, if and only if $f(\boldsymbol{a}) = g(\boldsymbol{a})$ for every element $\boldsymbol{a}$ of V^n.

Theorem 5 (Representation) Any fuzzy logic function f is expressed by

$$f(\boldsymbol{x}) = \bigvee_{a \in V_3^n} \left(\bigwedge_{a \succeq b} f(\boldsymbol{b}) \alpha^a(\boldsymbol{x}) \vee f(\boldsymbol{a}) \beta^a(\boldsymbol{x}) \right),$$

where α^a is a simple term corresponding to $\boldsymbol{a}$, and β^a is a complementary term corresponding to $\boldsymbol{a}$.

Proposition 1 Let α^a be a simple term corresponding to $\boldsymbol{a} \in V_3^n$.

(1) $\boldsymbol{a} \succeq \boldsymbol{b}$ if and only if $\alpha^a(\boldsymbol{b}) = 1$.

(2) If $\boldsymbol{b} \succ \boldsymbol{a}$, then $\alpha^a(\boldsymbol{b}) = 0.5$.

Proposition 2 For any $\boldsymbol{a}$ in V^n and any λ, τ in V, $\lambda \succ \tau$ implies $\overline{\boldsymbol{a}}^\tau \succ \overline{\boldsymbol{a}}^\lambda$.

3 Knowledge Discovery Based on Fuzzy Logic Functions

3.1 Problem Description

The problem of knowledge discovery is to acquire a general concept from specific training examples. The general concept to be retrieved can be formulated as a Boolean–valued function defined over some attributes, for which training examples are evaluated with true or false. Consider an example of knowledge "days good for diving," which depends on several possible attributes; *sunny*, *rainy*, *warm*, *windy*, and so on. Given a sample of days on which either good or bad is evaluated, we can make guess the notion of good–days–for–diving, e.g., *sunny* and *warm* or *not windy*.

In the real world, there actually is a day on which we can not determine whether it is a good for diving or not. Attributes might not be assigned to either true or false. The problem of knowledge discovery, therefore, should be able to deal with such fuzziness, and accordingly, generalizing Boolean–valued function, we have a fuzzy knowledge discovery based on fuzzy logic functions.

A fuzzy logic function is a mapping specified by a logic formula consisting of n variables and logical operations. Fundamental properties of fuzzy logic function have been studied and can be used to develop an effective algorithm to identify an optimal solution.

The training examples, however, might be inconsistent; there is a chance that training examples contain errors or noises, which will imply no solution. The knowledge discovery algorithm, therefore, should be able to detect the inconsistency in training examples, and preferably, find the optimal solution which minimizes errors even if an ill–defined training example is given. Thus, our objective in this section can be set as follows.

> **Knowledge Discovery:** given a set of training examples of target knowledge, finding the logic formulas that best fit the training examples.

To illustrate what we should attempt to solve, consider a set of training examples defined by a two–variable mapping h;

$$\begin{aligned} h(0.8, 0.7) &= 0.9, \\ h(0.3, 0.1) &= 0.6, \\ h(0.4, 0.8) &= 0.3, \\ h(0.7, 0.6) &= 0.8, \\ h(0.6, 0.9) &= 0.6. \end{aligned}$$

The value of h indicates a membership value to a target knowledge, for each attribute, x, y, being specific values. Given a training example, now we can test any logic formula, say $f = \neg x \wedge y$, whose evaluation for $(0.8, 0.7)$ is given as

$$f(0.8, 0.7) = \neg 0.8 \wedge 0.7 = \min(1 - 0.8, 0.7) = 0.2 \neq 0.9,$$

and the logic formula f turns out to be not any appropriate candidate for the target knowledge.

More formally, we give a definition of preferability of knowledge with a *distance* between two functions. Let F and G be mappings $F : A \to V$ and $G : B \to V$. The *distance* $E(F, G)$ between F and G is defined as

$$E(F, G) = \sum_{a \in A \cap B} (F(\boldsymbol{a}) - G(\boldsymbol{a}))^2.$$

The optimal solution to our problem is a fuzzy logic function, f^*, that minimizes the distance to the training examples, h.

The brute–force approach, in which testing all possible logic formula one by one, is infeasible because the number of fuzzy logic function asymptotically increases in $O(2^{3^n})$ with n being number of variables.

Our proposed algorithm is partitioned into several steps; (1) dividing the problem into some meaningful independent sub–problems, what we call *Q–equivalent classes*, and (2) we solve each sub–problems computing distance to the given h, and (3) then, combine local solutions to give the optimal fuzzy logic function which minimizes the total error against the training examples. Optionally, (4) we use a graph–theoretic algorithm to find f^*, which is a variation of Dijkstra's shortest path algorithm.

The proposed algorithm takes three steps; dividing, evaluating, and combining, described in this order.

3.2 Dividing into Q-Equivalent Classes

In this section, we introduce an equivalence relation, by which a number of training examples are classified into several small subsets.

Definition of Q-Equivalent Classes

Definition 7 Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ in V^n. A *quantized set* $C(\boldsymbol{a})$ of $\boldsymbol{a}$ is a subset of V_3^n defined by $C(\boldsymbol{a}) = \{\overline{\boldsymbol{a}}^\lambda \in V_3^n \mid \lambda \in V\}$.

Example 1 For the element $(0.4, 0.8)$, its quantized set is obtained below.

$$\begin{aligned} C((0.4, 0.8)) &= \{\overline{(0.4, 0.8)}^{0.4}, \overline{(0.4, 0.8)}^{0.8}, \overline{(0.4, 0.8)}^{1.0}\} \\ &= \{(0, 1), (0.5, 1), (0.5, 0.5)\}. \end{aligned}$$

Definition 8 Elements $\boldsymbol{a}$ and $\boldsymbol{b}$ of V^n are *Q-equivalent* if and only if the quantized sets satisfy $C(\boldsymbol{a}) = C(\boldsymbol{b})$. We write $\boldsymbol{a} \approx \boldsymbol{b}$ in this case.

Example 2 $(0.8, 0.7) \approx (0.7, 0.6)$, and $(0.3, 0.8) \approx (0.2, 0.9) \approx (0.4, 0.9)$.

Definition 9 Let A be a subset of V^n and $\boldsymbol{a} \in A$. A *Q-equivalent class* containing $\boldsymbol{a}$ is a subset of A defined by $[\boldsymbol{a}] = \{\boldsymbol{b} \in A \mid \boldsymbol{a} \approx \boldsymbol{b}\}$. The set of all Q-equivalent classes of A is denoted by $[A]$, i.e., $[A] = \{[\boldsymbol{a}] \subseteq A | \ \boldsymbol{a} \in A\}$.

Since $\approx$ is an equivalence relation, $[A] = \{A_1, \ldots, A_m\}$ is a partition of A, i.e., $A_i \cap A_j = \emptyset$ for every $i \neq j$, and $A = A_1 \cup \cdots \cup A_m$.

Example 3 Recalling the training examples h whose domain is D, we have the partition

$$\begin{aligned} D &= \{(0.4, 0.8), (0.6, 0.9), (0.7, 0.6), (0.8, 0.7), (0.3, 0.1)\} \\ &= \{(0.4, 0.8)\} \cup \{(0.6, 0.9)\} \cup \{(0.7, 0.6), (0.8, 0.7)\} \cup \{(0.3, 0.1)\} \\ &= D_1 \cup D_2 \cup D_3 \cup D_4. \end{aligned}$$

Figure 1 illustrates how we divide D into the Q–equivalent classes D_1, D_2, D_3 and D_4.

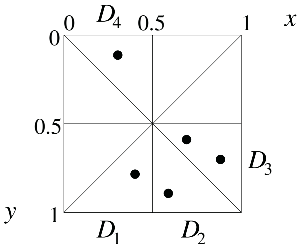


Fig. 1. Partition $[D]$: the set of training examples defined over $[0.1]^2$, indicated by dots, is partitioned into four Q–equivalent classes D_1, D_2, D_3, and D_4.

Some Properties of Q-Equivalent Classes. We clarify some properties of Q–equivalent classes that will be used to reduce complexity of searching.

Proposition 3 Let $C(\boldsymbol{a}) = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m\}$ be a Q–equivalent class. Then, $\boldsymbol{a}_i \succeq \boldsymbol{a}_j$ or $\boldsymbol{a}_j \succeq \boldsymbol{a}_i$ for every $i, j \in \{1, \ldots, m\}$. Note that this proposition means that there are indices $i_1 < i_2 < \cdots < i_m$ of $\{1, 2, \ldots, m\}$ such that $\boldsymbol{a}_{i_1} \succeq \boldsymbol{a}_{i_2} \succeq \cdots \succeq \boldsymbol{a}_{i_m}$.

Proposition 4 Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ be elements of V^n. Then, $\boldsymbol{a} \approx \boldsymbol{b}$ if and only if $\overline{a_i}^{a_j} = \overline{b_i}^{b_j}$ for every $i, j \in \{1, \ldots, n\}$.

Theorem 6 Let f and g be fuzzy logic functions, and $\boldsymbol{d} \in V^n$. Then, $f(\boldsymbol{a}) = g(\boldsymbol{a})$ for all $\boldsymbol{a} \in C(\boldsymbol{d})$ if and only if $f(\boldsymbol{b}) = g(\boldsymbol{b})$ for all $\boldsymbol{b} \in V^n$ such that $\boldsymbol{b} \approx \boldsymbol{d}$.

Proof. Suppose $f(\boldsymbol{b}) \neq g(\boldsymbol{b})$ for some $\boldsymbol{b} \in V^n$ when $f(\boldsymbol{a}) = g(\boldsymbol{a})$ for all $\boldsymbol{a} \in C(\boldsymbol{d})$. By Theorem 3, $\overline{f(\boldsymbol{b})}^{\lambda} = f(\overline{\boldsymbol{b}}^{\lambda}) \neq \overline{g(\boldsymbol{b})}^{\lambda} = g(\overline{\boldsymbol{b}}^{\lambda})$ for some $\lambda \in V$. This is contradictory to the hypothesis.

Conversely, suppose $f(\boldsymbol{a}) \neq g(\boldsymbol{a})$ for some $\boldsymbol{a} \approx \boldsymbol{d} \in C(\boldsymbol{d})$ when $f(\boldsymbol{b}) = g(\boldsymbol{b})$ for every $\boldsymbol{b} \approx \boldsymbol{d}$. Then, there is a $\boldsymbol{b} \in V^n$ such as $\overline{\boldsymbol{b}}^{\lambda} = \boldsymbol{a}$, and for $\boldsymbol{b}$, $f(\boldsymbol{a}) = f(\overline{\boldsymbol{b}}^{\lambda}) = \overline{f(\boldsymbol{b})}^{\lambda} \neq g(\boldsymbol{a}) = g(\overline{\boldsymbol{b}}^{\lambda}) = \overline{g(\boldsymbol{b})}^{\lambda}$. Since $\boldsymbol{b} \approx \boldsymbol{d}$, this contradicts the hypothesis. (End of Proof)

Example 4 For the element $(0.3, 0.8) \in V^2$, its quantized set is given as $C((0.3, 0.8)) = \{(0, 1), (0.5, 1), (0.5, 0.5)\}$. Then, Theorem 6 implies that for any fuzzy logic function f, the value $f(0.3, 0.8)$ can be evaluated uniquely if we have the values $f(0, 1)$, $f(0.5, 1)$ and $f(0.5, 0.5)$.

3.3 Evaluation of Q-Equivalent Classes

Restrictions of Fuzzy Logic Functions. According to Theorem 5, any fuzzy logic function f can be determined by 3^n parameters $f(\boldsymbol{a})$ for $\boldsymbol{a} \in V_3^n$. However, 3^n is too many to identify a fuzzy logic function. In this section, we will reduce the number of parameters.

Definition 10 Let $\boldsymbol{a} \in V_3^n$ and $b \in \{0,1\}$, and let f be a fuzzy logic function. Then, f is said to *be restricted to the truth status* $f(\boldsymbol{a}) = b$, if and only if $f(\boldsymbol{a}) = b$ and $f(\boldsymbol{c}) = 0.5$ for any $\boldsymbol{c} \in V_3^n$ such that $\boldsymbol{c} \succ \boldsymbol{a}$.

Note that, in general, there are some fuzzy logic functions f that are restricted to the same truth status $f(\boldsymbol{a}) = b$. For example, consider $f_1(x) = \neg x$ and $f_2(x) = x \vee \neg x$, both of which satisfy $f_1(0) = f_2(0) = 1$ and $f_1(0.5) = f_2(0.5) = 0.5$. Thus, f_1 and f_2 are restricted to $f_1(0) = f_2(0) = 1$. In the following discussion, we will use the symbol $f|_{f(a)=b}$ to mean any fuzzy logic function f that is restricted to $f(\boldsymbol{a}) = b$, and $f|_{f(a)=b}$ will be sometimes called a restriction.

Lemma 1 Let f be a fuzzy logic function and $\boldsymbol{d} \in V^n$. If $f(\boldsymbol{d}) \notin V_3$, then

(1) $f(\boldsymbol{a}) = 0.5$ for all $\boldsymbol{a} \in C(\boldsymbol{d})$ such that $\boldsymbol{a} \succ \boldsymbol{a}^*$,
(2) $f(\boldsymbol{a}^*) \in \{0,1\}$,

where $\boldsymbol{a}^* = \overline{\boldsymbol{d}}^{f(d)}$.

Proof. Since $f(\boldsymbol{d}) \notin V_3$, obviously, $\overline{f(\boldsymbol{d})}^{f(d)} = f(\overline{\boldsymbol{d}}^{f(d)}) = f(\boldsymbol{a}^*) \in \{0,1\}$. For any $\boldsymbol{b} \in V_3^n$ such that $\boldsymbol{b} \succ \boldsymbol{a}^*$, there is a $\lambda \in V$ such that $\boldsymbol{b} = \overline{\boldsymbol{d}}^{\lambda} \succeq \overline{\boldsymbol{d}}^{f(d)}$ and $f(\boldsymbol{d}) \succ \lambda$. Thus $\overline{f(\boldsymbol{d})}^{\lambda} = 0.5$, and thus, $f(\overline{\boldsymbol{d}}^{\lambda}) = f(\boldsymbol{b}) = 0.5$. (End of Proof)

Theorem 7 Let $\boldsymbol{d} \in V^n$ and f be a fuzzy logic function, and let $\boldsymbol{a}^* = \overline{\boldsymbol{d}}^{f(d)}$. Then, there exists a restriction $g|_{g(a^*)=f(a^*)}$ such that $f(\boldsymbol{c}) = g|_{g(a^*)=f(a^*)}(\boldsymbol{c})$ holds for every $\boldsymbol{c} \in V^n$ such that $\boldsymbol{c} \approx \boldsymbol{d}$.

Proof. For any $\boldsymbol{c} = \boldsymbol{a}^*$, obviously $g|_{g(a^*)=f(a^*)}(\boldsymbol{c}) = f(\boldsymbol{c})$. For any $\boldsymbol{c} \succeq \boldsymbol{a}^*$, $g|_{g(a^*)=f(a^*)}(\boldsymbol{c}) = 0.5 = f(\boldsymbol{c})$ [Lemma 1]. For any $\boldsymbol{a}^* \succeq \boldsymbol{c}$, $g|_{g(a^*)=f(a^*)}(\boldsymbol{c}) = f(\boldsymbol{a}^*) = f(\boldsymbol{c})$ [Theorem 2]. Thus, $g|_{g(a^*)=f(a^*)}(\boldsymbol{c}) = f(\boldsymbol{a}^*) = f(\boldsymbol{c})$ for all elements of $C(\boldsymbol{d})$, and by Theorem 6, $g|_{g(a^*)=f(a^*)}(\boldsymbol{c}) = f(\boldsymbol{a}^*) = f(\boldsymbol{c})$ for all elements of $[\boldsymbol{d}]$. (End of Proof)

This theorem implies that a fuzzy logic function can be determined uniquely within the given Q–equivalence class, and that for any fuzzy logic function f, there must be $f|_{f(a)=b}$ in a given Q–equivalence class. Hence, all combinations of $(\boldsymbol{a}, b)$ are sufficient to exhaustively enumerate all possible fuzzy logic functions.

Corollary 1 Let $D = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m\}$ be a subset of V^n, $D_1, \ldots, D_s$ be Q–equivalent classes of D, and F be any function $F : D \to V$. For any fuzzy logic function f, there are $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_s \in V_3^n$ and $b_1, \ldots, b_s \in \{0,1\}$ that satisfy

$$E(f,F) = \sum_{i=1}^{m} (f(\boldsymbol{d}_i) - F(\boldsymbol{d}_i))^2 = \sum_{i=1}^{s} E(f|_{f(a_i)=b_i}, F|_{D_i}),$$

where $F|_{D_i}$ is a mapping whose domain is restricted to D_i, i.e., $F|_{D_i} : D_i \to V$ and $F(\boldsymbol{a}) = F|_{D_i}(\boldsymbol{a})$ for any $\boldsymbol{a} \in D_i$.

Example 5 Let D_i be as in Example 3. For $f(x,y) = \neg y \vee (\neg x \wedge y)$,

$$\begin{aligned} E(f,h) = {} & E(f|_{f(0,1)=1}, h|_{D_1}) + E(f|_{f(1,0)=0}, h|_{D_2}) \\ & + E(f|_{f(1,0)=0}, h|_{D_3}) + E(f|_{f(0.5,0)=1}, h|_{D_4}). \end{aligned}$$

Representation of Restrictions. It has been guaranteed that any fuzzy logic function can be represented by combination of restrictions $f|_{f(a)=b}$. Then, we shift to a problem of representing each $f|_{f(a)=b}$ by a logic formula.

Theorem 8 Let $\boldsymbol{d} \in V^n$, $\boldsymbol{c} \in C(\boldsymbol{d})$, $b \in V_2$, and f be a fuzzy logic function. For $\boldsymbol{a} \in V^n$ such that $\boldsymbol{a} \approx \boldsymbol{d}$,

$$f|_{f(c)=b}(\boldsymbol{a}) = \begin{cases} \alpha^c(\boldsymbol{a}) & \text{if } b = 1 \\ \neg\alpha^c(\boldsymbol{a}) & \text{if } b = 0. \end{cases}$$

Proof. For $\boldsymbol{c} \succeq \boldsymbol{a}$, $f|_{f(c)=1}(\boldsymbol{a}) = 1 = \alpha^c(\boldsymbol{a}) = 1$,
and,
$f|_{f(c)=0}(\boldsymbol{a}) = 0 = \ \neg\alpha^c(\boldsymbol{a}) = 0$ [Proposition 1].
For $\boldsymbol{a} \succ \boldsymbol{c}$, $f|_{f(c)=b}(\boldsymbol{a}) = 0.5$
$= \alpha^c(\boldsymbol{a}) = \ \neg\alpha^c(\boldsymbol{a}) = 0.5$ [Proposition 1].
Thus, the theorem holds for all elements of $C(\boldsymbol{d})$.
Therefore, by Theorem 6,
$f|_{f(c)=1}(\boldsymbol{a}) = \alpha^c(\boldsymbol{a})$,
and,
$f|_{f(c)=0}(\boldsymbol{a}) = \ \neg\alpha^c(\boldsymbol{a})$ for every $\boldsymbol{a} \approx \boldsymbol{d}$. (End of Proof)

Example 6 Let $\boldsymbol{d} = (0.3, 0.8)$, and $\boldsymbol{c} = (0, 1) \in C(\boldsymbol{d})$. For any $\boldsymbol{a} \in V^n$ such that $\boldsymbol{a} \approx (0.3, 0.8)$,

$$\begin{aligned} f|_{f(0,1)=1}(\boldsymbol{a}) &= \alpha^{(0,1)}(\boldsymbol{a}) = (\neg x \wedge y)(\boldsymbol{a}), \\ f|_{f(0,1)=0}(\boldsymbol{a}) &= \neg\alpha^{(0,1)}(\boldsymbol{a}) = (\neg(\neg x \wedge y))(\boldsymbol{a}) = (x \vee \neg y)(\boldsymbol{a}). \end{aligned}$$

For any $(x, y) \in V^2$ such that $(x, y) \approx (0.3, 0.8)$,
$\overline{y}^x = \overline{0.8}^{0.3} = 1$ by Proposition 4. Thus, we have $1 \geq y \geq \max(x, 1 - x)$, and this implies the inequality $1 \geq y \geq \neg x \geq 0.5 \geq x \geq \neg y \geq 0$. Therefore, we have $f|_{f(0,1)=1}(\boldsymbol{x}) = \neg x$ and $f|_{f(0,1)=0}(\boldsymbol{x}) = x$.

Corollary 2 Let $[\boldsymbol{d}]$ be a Q-equivalent class of $\boldsymbol{d} \in V^n$, $\boldsymbol{a} \in C(\boldsymbol{d})$ and $b \in V_2$.

$$E(f|_{f(a)=b}, h|_{[d]}) = \begin{cases} \displaystyle\sum_{d' \in [d]} (h(\boldsymbol{d'}) - \alpha^a(\boldsymbol{d'}))^2 & \text{if } b = 1 \\ \displaystyle\sum_{d' \in [d]} (h(\boldsymbol{d'}) - \neg\alpha^a(\boldsymbol{d'}))^2 & \text{if } b = 0 \end{cases}$$

where $h|_{[d]}$ is a mapping whose domain is restricted to $[\boldsymbol{d}]$, i.e., $h|_{[d]} : [\boldsymbol{d}] \to V$ and $h|_{[d]}(\boldsymbol{a}) = h(\boldsymbol{a})$ for any $\boldsymbol{a} \in [\boldsymbol{d}]$.

Example 7 Recall h and D. We calculate every distance to h for each Q–equivalent class in $[D]$. For $\boldsymbol{d} = (0.8, 0.7)$ in D, we have the quantized set $C(\boldsymbol{d})$ and the Q-equivalent class $[\boldsymbol{d}]$:

$$\begin{aligned} C(\boldsymbol{d}) &= \{(1, 1), (1, 0.5), (0.5, 0.5)\}, \\ [\boldsymbol{d}] &= \{(0.8, 0.7), (0.7, 0.6)\}. \end{aligned}$$

Corollary 1 gives the distances for every fuzzy logic function by examining each $\boldsymbol{a}$ in $C(\boldsymbol{d})$, which involves $|C(\boldsymbol{d})| \cdot |V_2| = 3 \cdot 2 = 6$ restrictions in $[\boldsymbol{d}]$. Theorem 8 is useful to simplify the calculation of distances. For example,

$$\begin{aligned}
&E(f|_{f(1,1)=1}, h|_{[d]}) \\
&= \sum_{a \in [d]} (h(\boldsymbol{a}) - \alpha^{(1,1)}(\boldsymbol{a}))^2 \\
&= (h(0.8, 0.7) - (xy)(0.8, 0.7))^2 + (h(0.7, 0.6) - (xy)(0.7, 0.6))^2 \\
&= (0.9 - 0.7)^2 + (0.8 - 0.6)^2 = 0.08
\end{aligned}$$

$$E(f|_{f(1,1)=0}, h|_{[d]}) = (0.9 - 0.3)^2 + (0.8 - 0.4)^2 = 0.52$$

Table 1 lists the distances to h for each Q–equivalent class, $D_1, \ldots, D_4$.

Table 1. Distances $E(f, h)$.

$E(f|_{f(a)=b}, h|_{D_1})$

$\boldsymbol{a}\backslash b$	1	0
(0,1)	0.09	0.01
(0.5,1)	0.25	0.01
(0.5,0.5)	0.49	0.09

$E(f|_{f(a)=b}, h|_{D_2})$

$\boldsymbol{a}\backslash b$	1	0
(1,1)	0	0.04
(0.5,1)	0.09	0.04
(0.5,0.5)	0.16	0.36

$E(f|_{f(a)=b}, h|_{D_3})$

$\boldsymbol{a}\backslash b$	1	0
(1,1)	0.08	0.52
(1,0.5)	0.02	0.74
(0.5,0.5)	0.05	1.45

$E(f|_{f(a)=b}, h|_{D_4})$

$\boldsymbol{a}\backslash b$	1	0
(0,0)	0.01	0.09
(0.5,0)	0.09	0.25
(0.5,0.5)	0.14	0.36

3.4 Combining Local Solutions

In this subsection, we describe how we combine local solutions and find the global solution f^* with the shortest distance. In combining, there is an issue that not all combinations of restrictions can be fuzzy logic functions. In order to automatically determine conflicting classes, we show a necessary and sufficient condition for two distinct partially specified fuzzy logic functions to be combined consistently. Finally, we use a graph–theoretic algorithm that is a variation of Dijkstra's shortest path algorithm[12].

Condition for Two Functions to Be Consistent

Definition 11 Let A be a finite subset of V^n, and let F be a mapping $F : A \to V$. Then, subsets $C_i(F)$ and $C_i^*(F)$ of V_3^n defined below will be called *quantized sets* and *expansions* of F, respectively.

$$C_i(F) = \bigcup_{a \in A} \{\overline{\boldsymbol{a}}^\lambda \mid \exists \lambda \in V, \overline{F(\boldsymbol{a})}^\lambda = i\} \ (i = 0, 0.5, 1)$$

$$C_i^*(F) = C_i(F) \cup \bigcup_{b \in C_i(F)} \{\boldsymbol{a} \in V_3^n \mid \boldsymbol{b} \succeq \boldsymbol{a}\} \ (i = 0, 1)$$

$$C_{0.5}^*(F) = C_{0.5}(F) \cup \bigcup_{b \in C_1^*(F)} \{\boldsymbol{a} \in V_3^n \mid \boldsymbol{a} \succ \boldsymbol{b}\} \cup$$

$$\bigcup_{b \in C_0^*(F)} \{\boldsymbol{a} \in V_3^n \mid \boldsymbol{a} \succ \boldsymbol{b}\}$$

Example 8 Consider the function F such that $F(0.7, 0.5) = 0.3$ and $F(0.0, 1.0) = 0.8$.

We then have the quantized sets of F below:

$$\begin{aligned} C_0(F) &= \{\overline{(0.7, 0.5)}^{0.3}\} = \{(1, 0.5)\}, \\ C_1(F) &= \{(0, 1)\}, \text{ and} \\ C_{0.5}(F) &= \{\overline{(0.7, 0.5)}^{0.2}\} \cup \{(0, 1)\} = \{(0, 1), (0.5, 0.5)\}. \end{aligned}$$

Further, the expansions of F are given as

$$\begin{aligned} C_0^*(F) &= \{(1, 0.5), (1, 0), (1, 1)\} \\ C_1^*(F) &= \{(0, 1)\}, \text{ and} \\ C_{0.5}^*(F) &= \{(0, 1), (0.5, 0.5), (0.5, 1), (0, 0, 5)\}. \end{aligned}$$

Theorem 9 (Consistency) [14] Let A be a finite subset of V^n, and F be a mapping $f : A \rightarrow V$. There is a fuzzy logic function f such that $f(\boldsymbol{a}) = F(\boldsymbol{a})$ for all $\boldsymbol{a}$ in A, if and only if F satisfies both of the following conditions:

(R) $C_i^*(F) \cap C_j^*(F) = \emptyset$ for every $i \neq j$ $(i, j = 0, 0.5, 1)$, and
(N) $C_{0.5}(F) \cap V_2^n = \emptyset$.

Corollary 3 Let f and g be fuzzy logic functions, and let A and B be arbitrary finite subsets of V^n. Then, there is a fuzzy logic function f' that satisfies $f'(\boldsymbol{a}) = f|_A(\boldsymbol{a})$ for all $\boldsymbol{a}$ in A and $f'(\boldsymbol{b}) = g|_B(\boldsymbol{b})$ for all $\boldsymbol{b}$ in B, if and only if for every $i \neq j$ $(i, j = 0, 0.5, 1)$,

(R) $(C_i^*(f|_A) \cup C_i^*(g|_B)) \cap (C_j^*(f|_A) \cup C_j^*(g|_B)) = \emptyset$,

where $f|_A$ and $g|_B$ are mappings such that $f|_A : A \rightarrow V$ and $f(\boldsymbol{a}) = f|_A(\boldsymbol{a})$ for all $\boldsymbol{a} \in A$, and $g|_B : B \rightarrow V$ and $g(\boldsymbol{a}) = g|_B(\boldsymbol{a})$ for all $\boldsymbol{a} \in B$, respectively.

Example 9 For the following restrictions of fuzzy logic functions f_1 and f_2, $f_1(0, 1) = 1$, $f_1(0.5, 1) = 1$, $f_1(0.5, 0.5) = 0.5$, and $f_2(1, 1) = 0$, $f_2(1, 0.5) = 0$, $f_2(0.5, 0.5) = 0.5$, there is no fuzzy logic function that satisfies both f_1 and f_2. This is because the element $(1, 1) \in C_1^*(f_1) \cap C_0^*(f_2)$ violates the condition (R) in Corollary 3.

Replacement with Undirected Graph *G*

Definition 12 Let D be a finite subset of V^n. An (undirected) *graph* G characterized by D consists of two non–empty sets, the set $V(G)$ of vertices of G, and the set $Ed(G)$ of edges of G,

$$V(G) = \{f|_{[d]} \mid f \in \mathcal{F}_n, \boldsymbol{d} \in D\} = Ve_1 \cup Ve_2 \cup \cdots \cup Ve_s,$$

where $\mathcal{F}$ is a set of all n-variable fuzzy logic functions and Ve_i $(i = 1, \ldots, s)$ is defined by

$$Ve_i = \{f|_{[d_i]} \mid f \in \mathcal{F}_n\} = \{v_{i1}, v_{i2}, \ldots, v_{i(2(n+1))}\}.$$

Any edge $\{u, v\}$ in $Ed(G)$ satisfies the following condition:

(R) $(C_i^*(u) \cup C_i^*(v)) \cap (C_j^*(u) \cup C_j^*(v)) = \emptyset$ for every $i \neq j$, and
(A) there is no vertex w in $V(G)$ with edges $\{u, w\}$ and $\{w, v\}$.

A *weight* $W(v)$ of a vertex v in $V(G)$ is defined by

$$W(v) = E(v, h) = \sum_{d \in C(v)} (h(\boldsymbol{d}) - v(\boldsymbol{d}))^2,$$

where $E(v, h)$ is the distance between v and h, and $C(v)$ is the quantized set of v.

Example 10 Figure 2 illustrates the undirected graph G given by D in Example 3, where restrictions v_{ij} are indicated on the truth tables for V_3^2.

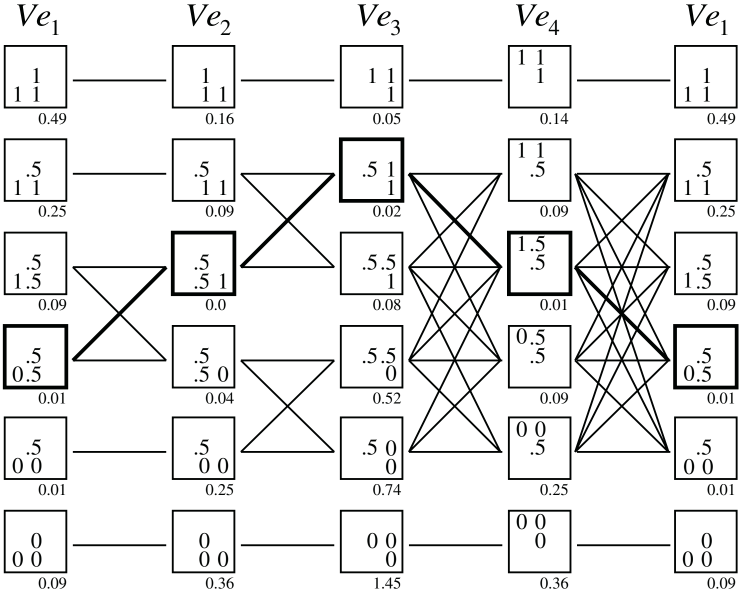


Fig. 2. Graph G: A set of restrictions in Q–equivalent classes are listed in a column, specified with boxes in which truth values are partially assigned. Distances for the training examples are indicated at the bottom of boxes. Consistent restrictions that satisfy condition to be combined are linked to each other. At the end of SPA algorithm, the shortest path from the third box from the bottom in the leftmost column to the rightmost column is given, as indicated with the bold links.

Shortest Path Algorithm(SPA). To find f^* with the shortest distance $E(f^*, h)$ is to find sequence $v_1^*, \ldots, v_s^*$ in $V(G)$ such that $\sum_{i=1}^{s} E(v_i^*, h)$ is the lowest value for all others. We prefer a quicker way without checking all possible combinations.

Assume $Ve_i, Ve_{i+1} \in V(G)$ are in order so that edge $\{v_i, v_{i+1}\} \in Ed(G)$ for some $v_i \in Ve_i$ and $v_{i+1} \in Ve_{i+1}$. Figure 3 shows an algorithm for shortest path algorithm that works with min–weight W_i.

```
For each $v$ in $Ve_1$
        Set $W^1(v) = E(v, h)$
        Set $P^1(v) = \{v\}$
End for
For $i = 2$ to $s$
        For each $v$ in $Ve_i$
                For each $u$ in $Ve_{i-1}$ with $\{u, v\}$ in $Ed(G)$
                        If $W^i(v) > W^{i-1}(u) + E(v, h)$
                                Set $W^i(v) = W^{i-1}(u) + E(v, h)$
                                Set $u^* = u$
                End for
                Set $P^i(v) = P^{i-1}(u^*) \cup \{v\}$
        End for
End for
```

Fig. 3. Shortest Path Algorithm(SPA).

Example 11 We apply the shortest path algorithm to the graph G in Example 10 (in Table 2).

Table 2. Matrices $W(G)$.

$j \backslash i$	1
1	.49
2	.25
3	.09
4	.01
5	.01
6	.09

$W^1(v_{ij})$

$j \backslash i$	1	2
1	.49	.65
2	.25	.34
3	.09	.01
4	.01	.05
5	.01	.26
6	.09	.45

$W^2(v_{ij})$

$j \backslash i$	1	2	3
1	.49	.65	.21
2	.25	.34	.02
3	.09	.01	.09
4	.01	.05	1.02
5	.01	.26	1.24
6	.09	.45	1.9

$W^3(v_{ij})$

$j \backslash i$	1	2	3	4
1	.49	.65	.21	.35
2	.25	.34	**.02**	.11
3	.09	**.01**	.09	**.03**
4	**.01**	.05	1.02	.11
5	.01	.26	1.24	.34
6	.09	.45	1.9	2.26

$W^4(v_{ij})$

Clearly the shortest path algorithm always stops, and the minimum weight in final W^s,

$$W^* = \min_{v \in Ve_s} W^s(v) = E(f^*, h)$$

is the intended distance to f^* and $P^* = P^s(v^*) = \{v_1^*, v_2*, \ldots, v_s^*\}$ gives f^*, that is, for any $\boldsymbol{d} \in D$,

$$f^*(\boldsymbol{d}) = v^*(\boldsymbol{d}) \quad \text{for some } v^* \in P^*.$$

Example 12 Given W^i and P^i in Example 11, we have

$$\begin{aligned} W^* &= \min_{v \in Ve_4} W^4(v) = W^4(v_{43}) = 0.03, \\ P^* &= \{v_{14}, v_{23}, v_{32}, v_{43}\}. \end{aligned}$$

According to Theorem 5, we can reconstruct the logic formula representing f^* as

$$\begin{aligned} f^*(x, y) &= xy \vee x \vee x\neg y \vee \neg x \neg y x \neg x y \vee x \neg x \neg y \vee \neg x y \neg y \\ &= x \vee \neg x \neg y. \end{aligned}$$

This is the answer f^* and we have solved the data–mining problem. Note that $x \vee \neg x \neg y \neq x \vee \neg y$ in fuzzy logic.

3.5 Estimation of SPA

If we simply examine fuzzy logic functions one by one, using a brute–force algorithm (BFA), it takes as much time as the number of n–variable fuzzy logic functions $|\mathcal{F}_n|$ that is greater than the number of n–variable Boolean logic functions $|\mathcal{B}_n| = 2^{2^n}$. The time to execute BFA is thus $O(2^{2^n})$.

For simplifying the estimation, we assume $D \in (V - V_3)^n$. At the dividing steps in Proposition 4, the permutation of n variables with two literals x_i and $\neg x_i$ is involved and leads to the number of Q–equivalent classes $|[D]|(= s)$ at most $2^n n!$, which can be substituted by m, the number of training examples because there are possibly some examples belonging to the common equivalent class.

For each Q–equivalent class $[\boldsymbol{d}]$, there are $2(n+1)$ restrictions of fuzzy logic functions. The number of vertices $|V(G)|$ is thus less than or equal to $2(n+1)m$.

The running time of the shortest path algorithm (SPA) is thus $O(nm)$.

The comparison/replacement step inside the u–loop takes at most some fixed amount of time. The step is done at most $|Ve_{i-1}| = 2(n+1)$ times in the u–loop, which is done $|Ve_i| = 2(n+1)$ times for the v–loop, which is done $2(n+1)m-1$ times for the outside i-loop. The total time to execute the algorithm is $(2(n+1))^2 \cdot 2(n+1)m = O(n^2 + nm)$.

4 Classification Method Based on Fuzzy Logic Functions

This section will describe an application of fuzzy logic functions to classification problems, i.e., a method for selecting a fuzzy logic function with a small number of product terms that enables us to classify as many as possible data in a numerical data set. This is one of the combinatorial problems with two objectives: to maximize the number of correctly classified objects and to minimize the number of product terms. Genetic algorithms will be applied to solve the problem.

4.1 Preliminaries

A typical example of a data set this section is focusing on is the Iris data set of Fisher. Table 3 shows some of examples of the Iris data set. In the data set, there are four attributes (sepal length, sepal width, petal length and petal width) and three classes (setosa, versicolor and virginica).

Table 3. Examples of Iris Plant Data Set.

x_1	x_2	x_3	x_4	Class
5.1	3.5	1.4	0.2	Setosa
4.9	3.0	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
6.3	3.3	6.0	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3.0	5.9	2.1	Virginica

x_1: sepal length in cm x_3: petal length in cm
x_2: sepal width in cm x_4: petal width in cm

Some assumptions on (numerical) data sets will be described below. A data set is assumed to have n attributes, $x_1, x_2, \ldots, x_n$ say; and m objects (patterns or training examples), $\boldsymbol{x}_1 = (x_1^1, \ldots, x_n^1)$, $\ldots$, $\boldsymbol{x}_m = (x_1^m, \ldots, x_n^m)$. For every object $\boldsymbol{x}_i = (x_1^i, \ldots, x_n^i)$, all elements x_j^i $(j = 1, \ldots, n)$ are assumed to be elements of the closed interval $[0,1]$, i.e., $\boldsymbol{x}_i$ is an element of $[0,1]^n$. So, the pattern space is defined by $[0,1]^n$. Furthermore, each object $\boldsymbol{x}_i$ is assumed to belong to one of p classes: Class 1, Class 2, $\ldots$, Class p. That is, the classification of each object is known as one of the p classes. Let Ω be a set of all objects (i.e., Ω is a finite subset of $[0,1]^n$), then every data set can be expressed by an incompletely specified function F from Ω into the set of classes $\{1, 2, \ldots, p\}$. This enables us to express a data set by the following p distinct incompletely specified functions F_i $(i = 1, 2, \ldots, p)$:

$$F_i(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_n) \text{ is Class } i \\ 0 & \text{otherwise,} \end{cases}$$

where $(x_1, \ldots, x_n)$ is an element of Ω.

Next, a classification method based on fuzzy logic functions will be briefly described. Let us assume that each axis of the pattern space $[0,1]^n$ is partitioned into k fuzzy subsets $\{A_1, A_2, \ldots, A_k\}$, where A_j is the j–th fuzzy subset. Any type of membership function (e.g., triangular, trapezoid, or exponential) can be employed to characterize the fuzzy subsets A_j's. For the sake of simplicity, this section will employ the following symmetric triangular membership functions:

For all $x \in [0,1]$,

$$\mu_j(x) = \max\left(1 - (k-1) \cdot \left| x - \frac{j-1}{k-1} \right|, \ 0\right) \quad (j = 1, 2, \ldots, k). \tag{2}$$

Now, let us introduce fuzzy logic functions based on the fuzzy partition. Let A_0 be a special type of fuzzy subset whose membership function μ_0 is characterized by the constant function 1, i.e., $\mu_0(x) = 1$ for all $x \in [0,1]$. Then, a fuzzy logic function f based on the fuzzy partition $\{A_1, A_2, \ldots, A_k\}$ (or simply, a fuzzy logic function f) is defined by the following sum–of–product form:

$$\begin{aligned} f &= \alpha_1 \vee \alpha_2 \vee \cdots \vee \alpha_m, \\ \alpha_t &= x_1^{A_t^1} x_2^{A_t^2} \cdots x_n^{A_t^n}, \end{aligned} \tag{3}$$

where α_t $(t = 1, 2, \ldots, m)$ is a product term; A_t^i $(i = 1, \ldots, n)$ is one of the fuzzy subsets $A_0, A_1, \ldots, A_k$; and $x_i^{A_t^i}$ stands for its membership function (i.e., $x_i^{A_t^i} = \mu_j(x_i)$ when A_t^i is equal to A_j). The logical operations $\vee$ and $\wedge$ follow Eq. (1) in Section 2.

Note that if a product term α contains a literal [1] $x_i^{A_0}$, then the attribute x_i has no effect on the product term α, i.e., the attribute x_i is independent from the product term α.

The aim of this section is to select a fuzzy logic function that is able to classify as many as possible objects in a data set. In this section we will try to solve the problem by finding p fuzzy logic functions of the form of Eq. (3) that are approximately expressing the p incompletely specified functions F_i $(i = 1, 2, \ldots, p)$, respectively. A classification system based on fuzzy logic functions is a collection of such p fuzzy logic functions. It is worth to notice that the number of product terms of a selected fuzzy logic function should be as small as possible because a fuzzy logic function with a small number of product terms gives us a compact classification system. So, the problem of this section can be formalized as the following combinatorial optimization problem with two objectives:

Problem A: For each $i = 1, 2, \ldots, p$, minimize

$$\mathit{error}(f_i) = \sum_{(x_1,\ldots,x_n)\in\Omega} (F_i(x_1, \ldots, x_n) - f_i(x_1, \ldots, x_n))^2$$

and minimize $\mathit{NP}(f_i)$, subject to $f_i \in \mathcal{F}_{\mathit{all}}$

In Problem A, $\mathit{NP}(f_i)$ denotes the number of product terms existing in a fuzzy logic function f_i; and $\mathcal{F}_{\mathit{all}}$ is a set of all fuzzy logic functions in the form of Eq. (3).

[1] Ordinarily, any positive variable x_i and its negation $\neg x_i$ are called literals. However, in this section, literals stand for the membership functions $x_i^{A_j}$ $(i = 1, \ldots, n$ and $j = 0, 1, \ldots, k)$.

4.2 An Application of Genetic Algorithm

A genetic algorithm will be applied to solve Problem A. First, this section will present a fitness function of the genetic algorithm and a way for encoding fuzzy logic functions into binary sequences. Then, its operations such as crossover, etc., will be shown.

Fitness Function and Coding. Because Problem A has two objectives, it is difficult to apply a genetic algorithm to the problem. So, Problem A is modified into the following problem with one objective by introducing a positive weight w.

Problem B: For each $i = 1, 2, \ldots, p$, minimize

$$\mathit{fitness}(f_i) = \mathit{error}(f_i) + w \cdot \mathit{NP}(f_i), \tag{4}$$

subject to $f_i \in \mathcal{F}_{all}$.

Since the high classification rate of a classification system is more required than its size, the weight w in Problem B should be set in order to satisfy the inequality $e(f_i) \gg w \cdot \mathit{NP}(f_i)$. We employ Eq. (4) as a fitness function of the genetic algorithm described below.

In the genetic algorithm, every fuzzy logic function in the form of Eq. (3) will be an individual from a population, and no individual beside that form exists. So, any sum–of–product form Eq. (3) will be directly represented by a binary sequence (a sequence, for short).

First, a way for representing a literal x^A by a sequence will be demonstrated. Given a set of fuzzy subsets $\{A_0, A_1, \ldots, A_k\}$, a literal x^{A_i} is encoded into the binary number corresponding to the subscript i of its fuzzy subset A_i. Table 4 shows the sequences of the literals x^{A_i} when $i = 0, 1, \ldots, 5$. Note that, in this case, the two sequences "110" and "111" are not permitted as codes of literals.

Table 4. Binary Codes for Literals $x^{A_0}, \ldots, x^{A_5}$.

Fuzzy Subset A	Literal x^A
A_0	000
A_1	001
A_2	010
A_3	011
A_4	100
A_5	101
–	110
–	111

Since a product term α has n distinct literals, its sequence can be defined by a collection of the sequences of the n literals. Furthermore, since a sum–of–product form Eq. (3) is a $\vee$–ing of some product terms, its sequence can also be defined by a collection of the sequences corresponding to the product terms.

Example 13 Consider the following fuzzy logic function in the form of Eq. (3).

$$f = x_1^{A_1} x_2^{A_3} x_3^{A_0} \vee x_1^{A_2} x_2^{A_0} x_3^{A_5},$$

where it is assumed that $n = 3$ and $k = 5$. Then, f can be encoded as follows.

001 011 000 - 010 000 101

The 9–bit sequence of the left half represents the first product term
$x_1^{A_1} x_2^{A_3} x_3^{A_0}$,
and the right half is corresponding to the second one.

Operations of Genetic Algorithm. The following genetic operations are employed to generate and handle a set of sequences (i.e., a population) in our genetic algorithm.

(1) *Initialization:* Generate an initial population with an appropriate number of sequences (i.e., fuzzy logic functions in the form of Eq. (3)). Let N_p be the number of sequences in the initial population. N_p may be an even.

(2) *Selection:* Select $N_p/2$ pairs of sequences from the current population. A roulette selection strategy is employed, i.e., a sequence s_i $(i = 1, 2, \ldots, N_p)$ is selected by the following selection probability P_s:

$$P_s(s_i) = \mathit{fitness}(s_i) / \sum_{j=1}^{N_p} \mathit{fitness}(s_j).$$

(3) *Crossover:* Perform a one–point crossover operation on each of the selected $N_p/2$ pairs of sequences with a crossover probability P_c. Note that every crossover point has to appear between two of neighbor literals in order to avoid producing a sequence that is not permitted as a code of a literal.

(4) *Mutation:* Perform the following two types of mutations.
   (A) For each literal, replace it with a literal randomly produced with a mutation probability P_m.
   (B) For each sequence, remove one of its product terms randomly selected with a mutation probability P_d. If no product term is removed, add a new product term randomly generated with a mutation probability P_a.

(5) *Elite Strategy:* Add the best sequence with the minimum fitness in the previous population to the current one. Then, remove the worst sequence from the current population.

(6) *Termination:* If a terminal condition is not satisfied, go to Step (2), stop the algorithm otherwise. If the algorithm meet the total number of generations N_g, then stop the algorithm.

4.3 Experiment

This section will describe the performance evaluation of classification systems based on fuzzy logic functions. The UCI (University of California at Irvine)–Machine–Learning Database will be applied in this evaluation, and we will show a comparison with some of other methods for classifying numerical data sets.

Data Sets, Evaluation Methods and Other Classification Systems. Five data sets of UCI–Machine–Learning Database are employed to evaluate the performance of classification systems based on fuzzy logic functions. Table 5 shows the number of objects, attributes and classes of the five data sets.

Table 5. Five Data Sets of UCI-Machine-Learning Database.

Data Set	Iris	Wine	Glass	Survival	Bupa
# Objects	150	178	214	306	345
# Attributes	4	13	9	3	6
# Classes	3	3	6	2	2

For estimating classification rate, this section considers the following three methods[16]:

(1) *Hold–Out (HO):* Leaning and testing on the whole data set.
(2) *Leave–One–Out (L1O):* Learning on the whole data set except the last, and testing the last. This procedure repeats until the number of permutations of objects.
(3) *Two–Fold–Cross–Validation (2F):* Dividing the data set at randomly into two parts, then learning on each part and testing on the other part.

Many classification methods were studied to classify numerical data sets. For comparing the performance of the method based on fuzzy logic functions with other methods, the following 8 methods are selected:

(1) Bayes Independent (Bayse)[15]
(2) Back Propagation Neural Network (BP)[13]
(3) Crisp k–Nearest Neighbor (CkNN)[1]
(4) Fuzzy k–Nearest Neighbor (FkNN)[9]
(5) Fuzzy Rules (FRule)[18]
(6) Fuzzy ID3 (FID3)[17]
(7) Fuzzy c–Means (FcM)[8]
(8) Hierarchical Fuzzy c–Means (HFcM)[16]

Note that in the rest of this section, the classification method based on fuzzy logic functions will be often referred as "FLogic".

Experimental Results. The genetic algorithm described in Section 4.2 was performed on the five data sets in Table 5. Its parameters were set to the following values, respectively, and the fuzzy subsets when $k = 5$ were applied.

# of Sequences:	$N_p = 20$
Crossover Probability:	$P_c = 0.8$
Mutation Probabilities:	$P_m = 0.05$
	$P_d = 0.05$
	$P_a = 0.05$
Total # of Generations:	$N_g = 50000$

The classification rates for FLogic are shown in Table 6 together with the other methods from Bayes to HFcM. Since HFcM can correctly classify all of the objects from a learning data set, its HO evaluation is ignored. FRule is space consuming because it has to produce all of fuzzy if–then rules, whose number is increasing exponentially with the number of attributes. So, some of its results could not be shown.

The following collection of three formulas is one of the results of FLogic when the Iris data set was applied to the genetic algorithm:

$$\begin{aligned} f_{\text{setosa}} &= x_3^{A_1} \vee x_4^{A_1} \vee x_1^{A_2} x_2^{A_4}, \\ f_{\text{versicolor}} &= x_3^{A_3} \vee x_4^{A_3}, \\ f_{\text{virginica}} &= x_4^{A_4} \vee x_4^{A_5}, \end{aligned}$$

where all of the literals with the fuzzy subset A_0 are omitted because they are the constant function 1. Here, this result can be replaced with the following fuzzy if–then rules.

If x_3 is A_1 then Class is Setosa
If x_4 is A_1 then Class is Setosa
If x_1 is A_2 and x_2 is A_4 then Class is Setosa
If x_3 is A_3 then Class is Versicolor
If x_4 is A_3 then Class is Versicolor
If x_4 is A_4 then Class is Virginica
If x_4 is A_5 then Class is Virginica

There are three methods that can produce fuzzy if–then rules; FRule, FID3 and FLogic. In Table 6, every number in parentheses indicates the number of fuzzy if–then rules. It is worth to notice that since FLogic, FRule and FID3 can be represented by a collection of fuzzy if–then rules, these methods are quite successful in extracting underlying structural information from a given data, which is useful in comprehending why and how phenomena occur.

By the results of Table 6, we can conclude that

(1) the classification rates of FLogic are not extremely high in comparing with the other classification methods. But, they are on the average or above it. Furthermore, FLogic is independent from evaluation methods such as HO, L1O and 2F, which means that FLogic can classify unknown objects as well as known objects.

(2) Bayes, FcM and HFcM do not correctly classify unknown objects because their classification rates are very low for L1O or 2F.

(3) for BP, CkNN and FkNN, many of their classification rates are more or less better than the rates of the other methods.

(4) for FID3, the number of fuzzy if–then rules depends on a terminal parameter. In this experiment, FID3 has made a large number of fuzzy if–then rules. FLogic and FRule, however, can produce a compact classification system (i.e., a system with a small number of fuzzy if–then rules).

(5) since FRule is space consuming, it did not work on three out of the five data sets.

Table 6. Classification Rates: This table shows the classification rates of test examples that can be correctly classified. FLogic stands for the method based on fuzzy logic functions discussed.

Methods	Evaluation	Iris	Wine	Glass	Survival	Bupa
Bayes	HO	0.95	0.99	0.68	0.77	0.66
	L1O	0.91	0.92	0.55	0.74	0.60
	2F	0.87	0.74	0.43	0.68	0.57
BP	HO	0.99	1.00	0.83	0.72	0.74
	L1O	0.94	0.98	0.64	0.50	0.63
	2F	0.95	0.98	0.57	0.64	0.64
CkNN	HO	0.96	0.98	0.82	0.78	0.79
	L1O	0.95	0.96	0.61	0.71	0.61
	2F	0.93	0.99	0.61	0.71	0.60
FkNN	HO	0.95	0.96	0.71	0.69	0.63
	L1O	0.92	0.95	0.70	0.57	0.61
	2F	0.92	0.99	0.61	0.69	0.60
FRule	HO	0.97(5)	over space	over space	0.81(18)	over space
	L1O	0.93(8)	over space	over space	0.75(18)	over space
	2F	0.89(10)	over space	over space	0.71(16)	over space
ID3	HO	0.95(19)	0.94(223)	0.83(628)	0.79(21)	0.73(321)
	L1O	0.95(18)	0.91(219)	0.65(430)	0.73(21)	0.61(318)
	2F	0.96(16)	0.92(107)	0.59(269)	0.74(21)	0.62(254)
FcM	HO	0.91	0.69	0.49	0.51	0.49
	L1O	0.92	0.64	0.37	0.42	0.49
	2F	0.89	0.67	0.46	0.49	0.48
HFcM	HO	–	–	–	–	–
	L1O	1.00	0.78	0.68	0.63	0.61
	2F	0.96	0.74	0.68	0.65	0.58
FLogic	HO	0.95(7)	0.90(9)	0.74(24)	0.76(21)	0.70(16)
	L1O	0.94(7)	0.91(9)	0.67(23)	0.72(19)	0.62(16)
	2F	0.94(6)	0.89(7)	0.74(21)	0.73(18)	0.61(16)

5 Conclusion

We have proposed a new method toward knowledge discovery with logic formulas and we have applied fuzzy logic functions to the classification problem.

We have proved that the division of the uncertain knowledge into some Q–equivalent classes is sufficient to calculate all possible local distances in Theorem 6, and we have shown some useful properties with respect to restrictions of fuzzy logic functions. The proposed algorithm finds the fuzzy logic function with the shortest distance to a given training examples in $O(n^2 + nm)$ time.

The experimental result shows that fuzzy logic functions gives results as accurate as some existing classification methods including Bayes, back–propagation, fuzzy k–nearest neighbor, fuzzy ID3, and fuzzy c–Means and its variation.

The acquired knowledge in our proposed method is a logic formula which gives human–readable representation of target knowledge.

References

1. T. M. Cover, P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory* **IT–13**, 1967, pp. 21–27.
2. R. C. T. Lee, C. L. Kang. Some properties of fuzzy logic. *Information and Control*, **19**, 1971, pp. 417–431.
3. A. Kandel. On minimization of fuzzy functions. *IEEE Trans. Computers* **22**, 1973, pp. 826–832.
4. M. Mukaidono. On some properties of fuzzy logic. *IECE Trans.* **58**, 1975, pp. 150–157 (in Japanese).
5. M. Mukaidono. An algebraic structure of fuzzy logic functions and its minimal and irredundant form. *IECE Trans.* **58**, 1975, pp. 748–755 (in Japanese).
6. M. Mukaidono. A necessary and sufficient condition for fuzzy logic functions, in: *Proceedings IEEE 9th International Symposium on Multiple–valued Logic*, 1979, pp. 159–166.
7. A. Kandel, J. M. Francioni. On the properties and applications of fuzzy–valued switching functions. *IEEE Trans. Computers* **29: 11**, 1980, pp. 986–994.
8. J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, 1981.
9. J. M. Keller, M. R. Gray, and J. A. Givens Jr. A fuzzy k–nearest neighbor algorithm. *IEEE Trans. Systems, Man, and Cybernetics* **SMC–15: 4**, 1985, pp. 580–585.
10. Liu Cu Hua. Minimization of fuzzy logic formula, in: *Proceedings IEEE 15th International Symposium on Multiple–Valued Logic*, 1985, pp. 182–189.
11. M. Mukaidono. The representation and minimization of fuzzy switching functions. *The Analysis of Fuzzy Information.* CRC Press, 1987, pp. 213–229.
12. K. A. Ross, C. R. B. Wright. *Discrete Mathematics.* Prentice-Hall, 1988.
13. D. E. Rumelhart et al. (eds.). *Parallel Distributed Processing.* The MIT Press, 1989.
14. H. Kikuchi. Necessary and sufficient conditions for restrictions to be fuzzy switching functions, in: *Proceedings of the International Fuzzy Engineering Symposium* **1**, 1991, pp. 91–102.
15. S. M. Weiss, C. A. Kulikowski. *Computer Systems That Learn.* Morgan Kaufmann Publishers,1991, pp. 51–79.
16. M. Grabisch, F. Dispot. A comparison of some methods of fuzzy classification on real data, in: *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks*, 1992, pp. 659–662.
17. M. Umano, H. Okamoto, I. Hatono, and H. Tamura. Fuzzy decision trees by fuzzy ID3 Algorithm and its application to diagnosis systems, in: *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, 1994, pp. 2113–2118.
18. H. Ishibuchi, K. Nozaki, and H. Tanaka. Selecting fuzzy if–then rules for classification problems using genetic algorithms. *IEEE Trans. Fuzzy systems* **3: 3**, 1995, pp. 260–270.
19. U. M. Fayyad et al. (eds.). *Advances in Knowledge Discovery and Data Mining.* AAAI Press/The MIT Press, 1996.

Fuzzy Integral Based Fuzzy Switching Functions

Eiichiro Takahagi

Senshu University, Kawasaki Kanagawa 214-8580, Japan

Abstract. This paper shows many relations among fuzzy switching functions and fuzzy integrals. First, we show that monotone fuzzy switching functions with constants are extended Sugeno integrals and fuzzy switching functions are Sugeno integrals whose inputs are double inputs, that is, true and false inputs. Next, we propose a Choquet integral based fuzzy binary switching function, which is calculated by a Choquet integral of double inputs and which uses adjusted fuzzy measure of complementary sets. The functions satisfy the complementary law and monotonicity under the monotonicity on binary inputs. Finally, we propose a logical Choquet integral and we discuss its properties.

1 Introduction

Fuzzy switching functions are used in many "fuzzy" fields, such as fuzzy inference, fuzzy control and so on. The fuzzy switching functions are not always monotone with respect to inputs in $[0, 1]$ even if it is monotone with respect to binary inputs. To decrease the effect, many researchers divide the input fields and make many fuzzy rules such as "If input 1 is negative small and input 2 is positive big then output is small".

In this paper, we propose a Choquet integral based fuzzy binary switching functions which satisfy the complementary law and are monotone under monotonicity on binary inputs.

Fuzzy switching functions and fuzzy integrals have many close relations. Marichal[1] pointed out that the Sugeno integral is a weighted max-min function and established mathematical properties of Sugeno integrals. Takahagi and Araki[2] discussed the relation among fuzzy switching functions with constants and Sugeno integrals and pointed out that they are transferable each other. In [3], Takahagi tried to use the Choquet integral instead of the Sugeno integral and proposed a primitive Choquet integral based fuzzy binary switching functions in [4].

2 Notations

2.1 Fuzzy Switching Function with Constants

Fuzzy switching functions with constraints (fuzzy/C, for short) are fuzzy switching functions that have constants, like,

$$f(x_1, x_2) = (0.6 \wedge x_1 \wedge \neg x_2) \vee (0.8 \wedge x_2) \vee 0.1. \tag{1}$$

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 129–150, 2004.

In equation (1), 0.7 , 0.5 and 0.1 are constants. In conventional fuzzy theories, that is max-min calculations, min operator is assigned to $\wedge$, max to $\vee$ and $1-x_i$ to $\neg x_i$. When we use those max-min calculations, we put superscript M. For example, when we calculate the value of equation (1) by max-min calculation:

$$f^M(x_1,x_2)=\max[\min(0.6,x_1,1-x_2),\min(0.8,x_2),0.1]. \tag{2}$$

Monotonicity of fuzzy/C is defined as,

$$x_1^1\geq x_1^2,\ldots,x_n^1\geq x_n^2\rightarrow f^M(x_1^1,\ldots,x_n^1)\geq f^M(x_1^2,\ldots,x_n^2). \tag{3}$$

2.2 Fuzzy Measures and Fuzzy Integrals

Let n be the number of inputs to a fuzzy integral, $X=\{1,\ldots,n\}$ be the set of inputs and 2^X be the set of all subsets of the X. Normally, a fuzzy measure μ is defined as,

$$\mu:2^X\rightarrow[0,1], \tag{4}$$

$$\mu(\emptyset)=0, \tag{5}$$

$$\mu(X)=1, \tag{6}$$

$$A\subseteq B\subseteq X\rightarrow\mu(A)\leq\mu(B). \tag{7}$$

Equation (5) is called “zero constraint of fuzzy measure”, equation (6) is called “normality of fuzzy measure” and equation (7) is called “monotonicity of fuzzy measure”. However, in this paper, we do not assume normality of fuzzy measure and in following sections, we relax the other conditions.

Let $h(i),i=1,\ldots,n$ be inputs to fuzzy integrals. The Sugeno integral is defined as,

$$\text{(S)}\int hd\mu\equiv\max_{r\in[0,1]}\min[r,\mu(\{x\mid h(x)>r\})], \tag{8}$$

and the Choquet integral is defined as,

$$\text{(C)}\int hd\mu\equiv\int_0^\infty\mu(\{x\mid h(x)>r\})dr. \tag{9}$$

In this paper, x_i, x_i^T and x_i^F are used to represent inputs of fuzzy/Cs while $h(i)$, $h(i^T)$ and $h(i^F)$ are used to represent inputs of fuzzy integrals.

3 Sugeno Integral Representations of Fuzzy/Cs

Fig. 1 is the outline of §3.1 and §3.2. Monotone fuzzy/Cs and Sugeno Integrals are transferable each others.

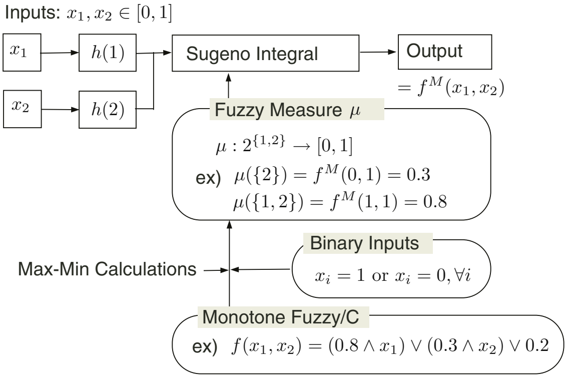

Inputs: $x_1, x_2 \in [0,1]$
x_1
x_2
$h(1)$
$h(2)$
Sugeno Integral
Output
$= f^M(x_1, x_2)$
Fuzzy Measure μ
$\mu : 2^{\{1,2\}} \to [0,1]$
ex) $\mu(\{2\}) = f^M(0,1) = 0.3$
$\mu(\{1,2\}) = f^M(1,1) = 0.8$
Binary Inputs
$x_i = 1$ or $x_i = 0, \forall i$
Max-Min Calculations
Monotone Fuzzy/C
ex) $f(x_1, x_2) = (0.8 \wedge x_1) \vee (0.3 \wedge x_2) \vee 0.2$

Fig. 1. Monotone Fuzzy/C to Sugeno Integral (2 Inputs Case).

3.1 Monotone and Zero Output Fuzzy/C

Weighted max-min function is defined as:

$$f^W(x_1, \ldots, x_n) = \bigvee_{T \subseteq X} [a_T \wedge (\bigwedge_{i \in T} x_i)]. \tag{10}$$

Marichal[1] pointed out that a Sugeno integrals is a weighted max-min function, where $x_i = h(i), i = 1, \ldots, n$ and $a_T = \mu(T)$.

A monotone fuzzy/C is defined as a fuzzy/C satisfying the monotonicity, i.e., equation (3). A zero output fuzzy/C is defined as a fuzzy/C satisfying,

$$f^M(0, \ldots, 0) = 0. \tag{11}$$

As a monotone and zero output Fuzzy/C is represented as a weighted max-min function, it is represented as a Sugeno integral.

Takahagi and Araki[2] point out that any monotone and zero output fuzzy/C and the Sugeno integral are transferable each other. Fuzzy measure μ is assigned from binary inputs of the fuzzy/C,

$$\begin{aligned} \mu(A) = f^M(x_1, \ldots, x_n) \quad &, \quad \forall A \in (2^X \setminus \emptyset) \\ x_i = 1 \quad &\text{if} \quad i \in A \\ x_i = 0 \quad &\text{if} \quad i \notin A \\ \mu(\emptyset) = 0 \quad &. \end{aligned} \tag{12}$$

By monotonicity of f^M, the fuzzy measure μ defined by (12) is also monotone. Using this fuzzy measure μ, the monotone and zero output fuzzy/C is represented as a Sugeno integral,

$$f^M(x_1, \ldots, x_n) = \text{(S)} \int h d\mu \ \text{ where } \ h(1) = x_1, \ldots, h(n) = x_n \tag{13}$$

$x_i \in [0,1], i = 1, \ldots, n$.

3.2 Monotone Fuzzy/C

A monotone fuzzy/C f^M is also represented as,

$$f^M(x_1,\ldots,x_n) = (\mathrm{S})\int h d\mu \vee C \ \text{ where } \ h(1)=x_1,\ldots,h(n)=x_n, \tag{14}$$

where $x_i \in [0,1], i=1,\ldots,n$ and $C = f(0,\ldots,0)$. In this representation, the constant C is outside of the Sugeno integral. To take the C inside of the Sugeno integral, we relax the definition of fuzzy measure by replacing zero constraint requirement (equation (5)) with,

$$\mu(\emptyset) \in [0,1]. \tag{15}$$

The assignment of fuzzy measure values from a fuzzy/C is as follows,

$$\begin{aligned} \mu(A) = f^M(x_1,\ldots,x_n) \quad &, \quad \forall A \in 2^X \\ x_i = 1 \ \ &\text{if} \ \ i \in A \\ x_i = 0 \ \ &\text{if} \ \ i \notin A \end{aligned} \tag{16}$$

In the definition of Sugeno integral(equation (8)), the constraint $\mu(\emptyset)=0$ is not necessary. Hence, we apply the equation (15) and the function is called extended Sugeno integral,

$$(\mathrm{ES})\int h d\mu \equiv \max_{r\in[0,1]} \min[r, \mu(\{x \mid h(x) > r\})]. \tag{17}$$

A monotone fuzzy/C is represented as an extended Sugeno integral,

$$f^M(x_1,\ldots,x_n) = (\mathrm{ES})\int h d\mu \ \text{ where } \ h(1)=x_1,\ldots,h(n)=x_n \tag{18}$$

$x_i \in [0,1], i=1,\ldots,n$.

3.3 Numerical Example of Monotone Fuzzy/Cs

We transfer a monotone fuzzy/C (19) to an extended Sugeno integral representation.

$$f(x_1,x_2) = (0.8 \wedge x_1) \vee (0.3 \wedge x_2) \vee 0.2 \tag{19}$$

From equation (16),

$$\begin{aligned} \mu(\emptyset) = f(0,0) = 0.2, &\quad \mu(\{1\}) = f(1,0) = 0.8, \\ \mu(\{2\}) = f(0,1) = 0.3, &\quad \mu(\{1,2\}) = f(1,1) = 0.8. \end{aligned}$$

Hence, $\forall x_1, x_2 \in [0,1]$,

$$f^M(x_1,x_2) = (\mathrm{ES})\int h d\mu, \ \text{ where } h(1)=x_1, h(2)=x_2. \tag{20}$$

3.4 Fuzzy/C

As Sugeno integrals are monotone functions and fuzzy/Cs include non-monotone functions, non-monotone fuzzy/Cs can not be represented as Sugeno integrals in the above way. Therefore, we propose the double input model which divide one input to true and false inputs, such as "two wire system". Fig. 2 outlines this process.

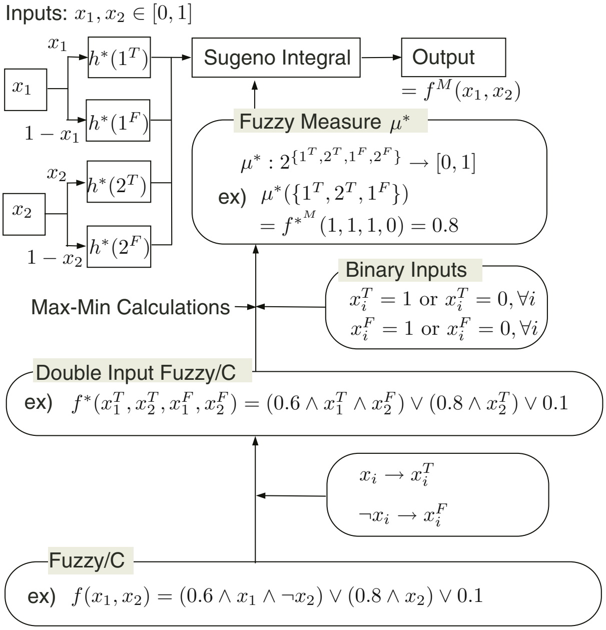

Inputs: $x_1, x_2 \in [0,1]$
x_1
$h^*(1^T)$
x_1
$h^*(1^F)$
$1 - x_1$
x_2
$h^*(2^T)$
x_2
$h^*(2^F)$
$1 - x_2$
Sugeno Integral
Output
$= f^M(x_1, x_2)$
Fuzzy Measure μ^*
$\mu^* : 2^{\{1^T,2^T,1^F,2^F\}} \to [0,1]$
ex) $\mu^*(\{1^T, 2^T, 1^F\})$
$= f^{*^M}(1,1,1,0) = 0.8$
Binary Inputs
$x_i^T = 1$ or $x_i^T = 0, \forall i$
$x_i^F = 1$ or $x_i^F = 0, \forall i$
Max-Min Calculations
Double Input Fuzzy/C
ex) $f^*(x_1^T, x_2^T, x_1^F, x_2^F) = (0.6 \wedge x_1^T \wedge x_2^F) \vee (0.8 \wedge x_2^T) \vee 0.1$
$x_i \to x_i^T$
$\neg x_i \to x_i^F$
Fuzzy/C
ex) $f(x_1, x_2) = (0.6 \wedge x_1 \wedge \neg x_2) \vee (0.8 \wedge x_2) \vee 0.1$

Fig. 2. Outline of Fuzzy/Cs to Sugeno Integral (2 Inputs Case).

In this model, we distinguish positive variables (x_i) and negative variables ($\neg x_i$) and replace x_i by x_i^T and $\neg x_i$ by x_i^F. Let f^* be the replaced formula and is called a double input fuzzy/C.

For example, f^* of equation (1) is:

$$f^*(x_1^T, x_2^T, x_1^F, x_2^F) = (0.6 \wedge x_1^T \wedge x_2^F) \vee (0.8 \wedge x_2^T) \vee 0.1. \tag{21}$$

In this representation, we deal with x_i^T and x_i^F as different variables:

$$f^* : [0,1]^{2n} \to [0,1]. \tag{22}$$

As f^* is a monotone function and from §3.2, f^* is represented as an extended Sugeno integral in,

$$X^* = \{1^T, \dots, n^T, 1^F, \dots, n^F\}, \tag{23}$$

$$h^* \ : \ X^* \to [0,1], \tag{24}$$

$$h(i^T) = x_i^T = x_i, \tag{25}$$

$$h(i^F) = x_i^F = 1 - x_i. \tag{26}$$

Fuzzy measure μ^* is assigned from f^*:

$$\mu^*(A) = f^{*M}(x_1^T, \ldots, x_n^T, x_1^F, \ldots, x_n^F), \qquad \forall A \in 2^{X^*} \tag{27}$$

$$\begin{aligned} x_i^T &= 1 \quad \text{if} \quad i^T \in A \\ x_i^T &= 0 \quad \text{if} \quad i^T \notin A \\ x_i^F &= 1 \quad \text{if} \quad i^F \in A \\ x_i^F &= 0 \quad \text{if} \quad i^F \notin A. \end{aligned}$$

Then, any fuzzy/C is represented as an extended Sugeno integral,

$$f^M(x_1, \ldots, x_n) = (\mathrm{ES}) \int h^* d\mu^* \ \text{ where } \ h^*(i^T) = x_i, h^*(i^F) = 1 - x_i, \tag{28}$$

$x_i \in [0,1], i = 1, \ldots, n.$

3.5 Numerical Example of Fuzzy/Cs

Equation (1) is a non-monotone fuzzy/C and f^* of equation (1) is equation (21). $n = 2$ and $X^* = \{1^T, 2^T, 1^F, 2^F\}$. From equation (27), we get μ^* shown in Table 1.

Table 1. μ^* of equation (21).

$\mu^*(\emptyset) = f^{*M}(0,0,0,0) = 0.1,$	$\mu^*(\{1^T\}) = f^{*M}(1,0,0,0) = 0.1,$
$\mu^*(\{2^T\}) = f^{*M}(0,1,0,0) = 0.8,$	$\mu^*(\{1^F\}) = f^{*M}(0,0,1,0) = 0.1,$
$\mu^*(\{2^F\}) = f^{*M}(0,0,1,0) = 0.1,$	$\mu^*(\{1^F, 2^F\}) = f^{*M}(0,0,1,1) = 0.1,$
$\mu^*(\{1^T, 2^T\}) = f^{*M}(1,1,0,0) = 0.8,$	$\mu^*(\{1^T, 1^F\}) = f^{*M}(1,0,1,0) = 0.1,$
$\mu^*(\{1^T, 2^F\}) = f^{*M}(1,0,0,1) = 0.6,$	$\mu^*(\{1^T, 2^T, 1^F\}) = f^{*M}(1,1,1,0) = 0.8,$
$\mu^*(\{2^T, 2^F\}) = f^{*M}(0,1,0,1) = 0.8,$	$\mu^*(\{1^T, 1^F, 2^F\}) = f^{*M}(1,1,1,0) = 0.6,$
$\mu^*(\{2^T, 1^F\}) = f^{*M}(0,1,1,0) = 0.8,$	$\mu^*(\{1^T, 2^T, 2^F\}) = f^{*M}(1,1,1,0) = 0.8,$
$\mu^*(X^*) = f^{*M}(1,1,1,1) = 0.8,$	$\mu^*(\{2^T, 1^F, 2^F\}) = f^{*M}(1,1,1,0) = 0.8,$

For example, $f^M(0.8, 0.4) = \max[\min(0.6, 0.8, 0.6), \min(0.8, 0.4), 0.1] = 0.6$. As $h^*(1^T) = 0.8, h^*(2^T) = 0.4, h^*(1^F) = 0.2, h^*(2^F) = 0.6$, the extended Sugeno integrated value is:

$$\begin{aligned} &(\mathrm{ES}) \int h^* d\mu^* \\ =& \max[\min(1.0, \mu(\emptyset)), \min(0.8, \mu(\{1^T\})), \min(0.6, \mu(\{1^T, 2^F\}), \\ & \min(0.4, \mu(\{1^T, 2^T, 2^F\}), \min(0.2, \mu(X^*)] \\ =& \max[\min(1, 0.1), \min(0.8, 0.1), \min(0.6, 0.6), \min(0.4, 0.8), \min(0.2, 0.8)] \\ =& 0.6. \end{aligned}$$

Table 2 is the conclusion of this section.

Table 2. Fuzzy/Cs and Sugeno integrals.

Fuzzy/Cs		Fuzzy measures
Monotone Fuzzy/C, $f^M(0,\ldots,0)=0$	$\Longleftrightarrow$	$\mu : 2^X \to [0,1]$, $\mu(\emptyset)=0$
Monotone Fuzzy/C	$\Longleftrightarrow$	$\mu : 2^X \to [0,1]$, $\mu(\emptyset)\in[0,1]$
Fuzzy/C	$\Rightarrow$	$\mu^* : 2^{X^*} \to [0,1]$, $\mu^*(\emptyset)\in[0,1]$

4 Choquet Integral Based Fuzzy Binary Switching Functions

Fuzzy/Cs do not satisfy the monotonicity even if they are monotone with respect to binary inputs. For example,

$$f(x_1) = (1.0 \wedge x_1) \vee (0.6 \wedge \neg x_1) \tag{29}$$

satisfies the monotonicity with respect to binary inputs, that is $f^M(1) \geq f^M(0)$. However, $f^M(x_1), x_1 \in [0,1]$ is not a monotone fuzzy/C. Fig. 3 shows that $f^M(x_1)$ decrease at $0.4 < x_1 < 0.5$.

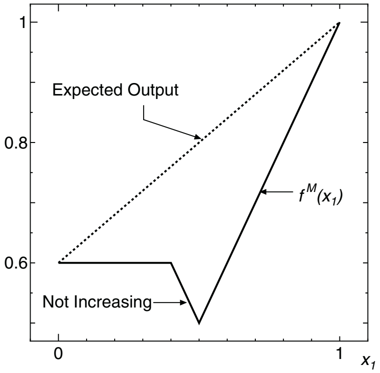


Fig. 3. Output of $f^M(x_1) = 1.0 \wedge x_1 \vee 0.6 \wedge \neg x_1$.

We think, equation (29) means that "output is 0.6 at $x_1 = 0$, output is 1.0 at $x_1 = 1$ and outputs at other $x_1 \in (0,1)$ are interpolated". If we stand on this statement, then outputs of $f(x_1)$ are the dotted line of Fig. 3. Choquet integral based Fuzzy Binary switching functions (CFB functions for short) output the dotted line which is monotone with respect to x_1. Generally, CFB functions satisfy the monotonicity if they are monotone with respect to binary inputs.

In this section, we propose the calculation methods of CFB functions and show the properties of the functions. Fig. 4 outlines the calculation process of a CFB function.

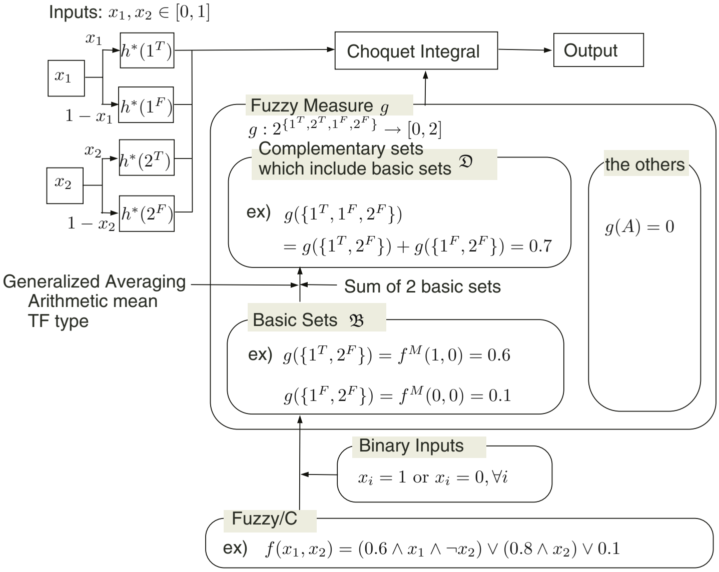

Inputs: $x_1, x_2 \in [0,1]$
x_1
x_1
$h^*(1^T)$
$1-x_1$
$h^*(1^F)$
x_2
x_2
$h^*(2^T)$
$1-x_2$
$h^*(2^F)$
Choquet Integral
Output
Fuzzy Measure g
$g : 2^{\{1^T,2^T,1^F,2^F\}} \to [0,2]$
Complementary sets which include basic sets $\mathfrak{D}$
ex) $g(\{1^T,1^F,2^F\})$
$= g(\{1^T,2^F\}) + g(\{1^F,2^F\}) = 0.7$
the others
$g(A) = 0$
Generalized Averaging
Arithmetic mean
TF type
Sum of 2 basic sets
Basic Sets $\mathfrak{B}$
ex) $g(\{1^T,2^F\}) = f^M(1,0) = 0.6$
$g(\{1^F,2^F\}) = f^M(0,0) = 0.1$
Binary Inputs
$x_i = 1$ or $x_i = 0, \forall i$
Fuzzy/C
ex) $f(x_1,x_2) = (0.6 \wedge x_1 \wedge \neg x_2) \vee (0.8 \wedge x_2) \vee 0.1$

Fig. 4. Outline Process of CFB Functions (2 Inputs Case).

4.1 Definitions of CFB Functions

CFB functions are calculated by Choquet integrals having double inputs,

$$X^* = \{1^T, \ldots, n^T, 1^F, \ldots, n^F\}, \tag{30}$$

$$h^* : X^* \to [0,1]. \tag{31}$$

Each input has true input i^T and false input i^F; i^T is called i^F's complementary element, and, dually, i^F is called i^T's complementary element. The pair (i^T, i^F) is called complementary pair. When a set A includes a complementary pair, A is called complementary set. $\mathfrak{C}$ is defined as the set of all complementary sets,

$$\mathfrak{C} \equiv \{A \mid \exists j, j^T \in A, j^F \in A, A \subseteq X^*\}. \tag{32}$$

$\mathfrak{N}$ is defined as the set of all non-complementary sets,

$$\mathfrak{N} \equiv 2^{X^*} \setminus \mathfrak{C}. \tag{33}$$

Basic set B is defined as the set which includes all number's elements and only one element for each number. An example of basic set is $\{1^T, 2^F, 3^F, 4^F, 5^T, 6^T\}$, $n = 6$. $\mathfrak{B}$ is the set of all basic sets of X^*,

$$\mathfrak{B} \equiv \{B \mid B \in \mathfrak{N}, \mid B \mid = n, B \subseteq X^*\}. \tag{34}$$

$\mathfrak{D}$ is defined as the set of complementary sets which include basic sets,

$$\mathfrak{D} \equiv \{D \mid D \in \mathfrak{C}, \exists B \in \mathfrak{B}, D \supset B, D \subseteq X^*\}. \tag{35}$$

4.2 Fuzzy Measure Assignment

Fuzzy measure's range of CFB functions is the interval $[0, 2]$,

$$g : 2^{X^*} \to [0, 2]. \tag{36}$$

Fuzzy measure values of non-complementary and non-basic sets are 0,

$$g(A) = 0, \;\; \forall A \in (\mathfrak{N} \setminus \mathfrak{B}). \tag{37}$$

Fuzzy measure values of complementary sets which do not include basic sets are also 0,

$$g(A) = 0, \;\; \forall A \in (\mathfrak{C} \setminus \mathfrak{D}). \tag{38}$$

Fuzzy measure values of basic sets are assigned from fuzzy/C's output values with respect to binary inputs:

$$\begin{aligned} g(B) = f^M(&x_1, \ldots, x_n) \\ &x_i = 1 \qquad \text{if } \; i^T \in B \\ &x_i = 0 \qquad \text{if } \; i^F \in B, \\ &\forall B \in \mathfrak{B}. \end{aligned} \tag{39}$$

Fuzzy measure values of $D \in \mathfrak{D}$ is defined as the sum of two fuzzy measure values of basic sets which cover the D, that is, $D = B_1 \cup B_2, B_1, B_2 \in \mathfrak{B}$, $g(D) = g(B_1) + g(B_2)$. The pair (B_1, B_2) is called a partition pattern. If D includes at least two complementary element pairs, then the partition patterns to the basic sets are at least two. In this case, $g(D)$ is the averaging value of all partition patterns,

$$g(D) = \underset{\{(B_1,B_2) | B_1 \cup B_2 = D, B_1, B_2 \in \mathfrak{B}\}}{\mathrm{GAV}} [g(B_1) + g(B_2)], \;\; \forall D \in \mathfrak{D} \tag{40}$$

where GAV is the generalized averaging function which satisfies the following two properties,

1. $\min(a_1, \ldots, a_k) \le \mathrm{GAV}(a_1, \ldots, a_k) \le \max(a_1, \ldots, a_k)$,
2. $\mathrm{GAV}(a_1, \ldots, a_k) = \mathrm{GAV}(a_1, \ldots, a_k, a_1, \ldots, a_k)$.

An example of GAV is the arithmetic mean:

$$g^{AM}(D) = \frac{\sum_{\{(B_1,B_2) | B_1 \cup B_2 = D, B_1, B_2 \in \mathfrak{B}\}} [g^{AM}(B_1) + g^{AM}(B_2)]}{| \{(B_1, B_2) \mid B_1 \cup B_2 = D, B_1, B_2 \in \mathfrak{B}\} |}, \;\; \forall D \in \mathfrak{D}, \tag{41}$$

where

$$g^{AM}(A) = g(A), \forall A \in (2^X \setminus \mathfrak{D}). \tag{42}$$

To reduce computational effort, we also use T-F type mean. T-F type mean is the sum of two particular basic sets B_T and B_F. B_T is the basic set which includes true elements of complementary elements of D and non-complementary elements of D. B_F is the basic set that includes false elements of complementary elements of D and non-complementary elements of D. That is,

$$C_D = \{i \mid i^T \in D, i^F \in D\}, \tag{43}$$
$$N_D = D \setminus \{i^T, i^F \mid i \in C_D\}, \tag{44}$$
$$B_T = \{i^T \mid i \in C_D\} \cup N_D, \tag{45}$$
$$B_F = \{i^F \mid i \in C_D\} \cup N_D. \tag{46}$$

$$g^{TF}(D) = g^{TF}(B_T) + g^{TF}(B_F), \forall D \in \mathfrak{D}, \tag{47}$$

where $g^{TF}(A) = g(A), \forall A \in (2^X \setminus \mathfrak{D})$.
For example, $n = 4$, $g^{TF}(\{1^T, 1^F, 2^T, 3^T, 3^F, 4^F\}) = g^{TF}(\{1^T, 2^T, 3^T, 4^F\}) + g^{TF}(\{1^F, 2^T, 3^F, 4^F\})$.

4.3 Definition of CFB Functions

CFB functions are defined as Choquet integral using the g:

$$f^{CFB(AM)}(x_1, \dots, x_n) \equiv \text{(C)} \int h^* dg^{AM}, \tag{48}$$
$$\text{where } h^*(i^T) = x_i, h^*(i^F) = 1 - x_i, \forall i$$

and

$$f^{CFB(TF)}(x_1, \dots, x_n) \equiv \text{(C)} \int h^* dg^{TF}, \tag{49}$$
$$\text{where } h^*(i^T) = x_i, h^*(i^F) = 1 - x_i, \forall i.$$

When we show the properties which hold for any GAV, we display f^{CFB}.

4.4 Numerical Examples of CFB Functions

A numerical example of a CFB function is equation (1); its input values are $x_1 = 0.8, x_2 = 0.4$.

$$\begin{aligned}
X^* =& \{1^T, 2^T, 1^F, 2^F\},\\
\mathfrak{C} =& \{\{1^T, 1^F\}, \{2^T, 2^F\}, \{1^T, 1^F, 2^T\}, \{1^T, 1^F, 2^F\}, \{1^T, 2^T, 2^F\},\\
& \{1^F, 2^T, 2^F\}, \{1^T, 2^T, 1^F, 2^F\}\},\\
\mathfrak{N} =& \{\emptyset, \{1^T\}, \{2^T\}, \{1^F\}, \{2^F\}, \{1^T, 2^T\}, \{1^T, 2^F\}, \{1^F, 2^T\}, \{1^F, 2^F\}\},\\
\mathfrak{B} =& \{\{1^T, 2^T\}, \{1^T, 2^F\}, \{1^F, 2^T\}, \{1^F, 2^F\}\},\\
\mathfrak{D} =& \{\{1^T, 1^F, 2^T\}, \{1^T, 1^F, 2^F\}, \{1^T, 2^T, 2^F\}, \{1^F, 2^T, 2^F\},\\
& \{1^T, 2^T, 1^F, 2^F\}\}
\end{aligned}$$

From equation (37),

$$g(\emptyset) = g(\{1^T\}) = g(\{2^T\}) = g(\{1^F\}) = g(\{2^F\}) = 0. \tag{50}$$

From equation (38),

$$g(\{1^T,1^F\}) = g(\{2^T,2^F\}) = 0. \tag{51}$$

From equation (39), fuzzy measure values of basic sets are assigned from binary inputs of the fuzzy/C,

$$g(\{1^T,2^T\}) = f^M(1,1) = 0.8, \tag{52}$$
$$g(\{1^T,2^F\}) = f^M(1,0) = 0.6, \tag{53}$$
$$g(\{1^F,2^T\}) = f^M(0,1) = 0.8, \tag{54}$$
$$g(\{1^F,2^F\}) = f^M(0,0) = 0.1. \tag{55}$$

As above fuzzy measure values do not depend on GAV function, $g^{AM}(A) = g^{TF}(A) = g(A), \forall A \in (2^{X^*} \setminus \mathfrak{D})$.

Fuzzy measure values of $\mathfrak{D}$ is assigned from equation (40). As $\{1^T,1^F,2^T\}$, $\{1^T,1^F,2^F\}$, $\{1^T,2^T,2^F\}$ and $\{1^F,2^T,2^F\}$ have only one partition pattern, fuzzy measure value calculation methods do not depend on GAV function,

$$g^{AM}(\{1^T,1^F,2^T\}) = g^{TF}(\{1^T,1^F,2^T\}) = g(\{1^T,2^T\}) + g(\{1^F,2^T\}) = 1.6,$$
$$g^{AM}(\{1^T,1^F,2^F\}) = g^{TF}(\{1^T,1^F,2^F\}) = g(\{1^T,2^F\}) + g(\{1^F,2^F\}) = 0.7,$$
$$g^{AM}(\{1^T,2^T,2^F\}) = g^{TF}(\{1^T,2^T,2^F\}) = g(\{1^T,2^T\}) + g(\{1^T,2^F\}) = 1.4,$$
$$g^{AM}(\{1^F,2^T,2^F\}) = g^{TF}(\{1^F,2^T,2^F\}) = g(\{1^F,2^T\}) + g(\{1^F,2^F\}) = 0.9.$$

As $\{1^T,2^T,1^F,2^F\}$ have two partition patterns, that are $(\{1^T,2^T\},\{1^F,2^F\})$ and $(\{1^T,2^F\},\{1^F,2^T\})$, fuzzy measure value calculation methods depend on GAV function chosen,

Arithmetic mean:

$$\begin{aligned} &g^{AM}(\{1^T,2^T,1^F,2^F\}) \\ =&\frac{[g(\{1^T,2^T\}) + g(\{1^F,2^F\})] + [g(\{1^T,2^F\}) + g(\{1^F,2^T\})]}{2} = 1.15, \end{aligned}$$

TF type:

$$g^{TF}(\{1^T,2^T,1^F,2^F\}) = g(\{1^T,2^T\}) + g(\{1^F,2^F\}) = 0.9.$$

The outputs to the CFB function are Choquet integrated values. As $x_1 = 0.8$ and $x_2 = 0.4$, $h^*(1^T) = 0.8$, $h^*(1^F) = 0.2$, $h^*(2^T) = 0.4$ and $h^*(2^F) = 0.6$. Therefore,

$$\begin{aligned} &f^{CFB(AM)}(0.8,0.4) = (\mathrm{C}) \int h^* dg^{AM} \\ =&[h^*(1^T) - h^*(2^F)]g^{AM}(\{1^T\}) + [h^*(2^F) - h^*(2^T)]g^{AM}(\{1^T,2^F\}) + \\ &[h^*(2^T) - h^*(1^F)]g^{AM}(\{1^T,2^T,2^F\}) + h^*(1^F)g^{AM}(\{1^T,1^F,2^T,2^F\}) \\ =&0.2 \times 0 + 0.2 \times 0.6 + 0.2 \times 1.4 + 0.2 \times 1.15 = 0.63 \end{aligned}$$

and

$$
\begin{aligned}
&f^{CFB(TF)}(0.8, 0.4) = \text{(C)} \int h^* dg^{TF} \\
=&[h^*(1^T) - h^*(2^F)]g^{TF}(\{1^T\}) + [h^*(2^F) - h^*(2^T)]g^{TF}(\{1^T, 2^F\}) + \\
&[h^*(2^T) - h^*(1^F)]g^{TF}(\{1^T, 2^T, 2^F\}) + h^*(1^F)g^{TF}(\{1^T, 1^F, 2^T, 2^F\}) \\
=&0.2 \times 0 + 0.2 \times 0.6 + 0.2 \times 1.4 + 0.2 \times 0.9 = 0.58.
\end{aligned}
$$

4.5 Extensions of Binary Switching Functions

CFB functions are extensions of binary switching functions, that is, outputs of CFB functions on binary inputs have same outputs as binary switching functions. Let f^B be a (binary) switching function, therefore,

$$f^{CFB}(x_1, \ldots, x_n) = f^B(x_1, \ldots, x_n), \forall x_i \in \{0, 1\}, i = 1, \ldots, n. \tag{56}$$

The property which holds for fuzzy/Cs on binary inputs also holds for CFB functions on binary inputs:

$$f^{CFB}(x_1, \ldots, x_n) = f^M(x_1, \ldots, x_n), \forall x_i \in \{0, 1\}, i = 1, \ldots, n. \tag{57}$$

4.6 Complementary Law

If two fuzzy/Cs have same output values on all binary input values, then the two CFB functions of the fuzzy/C are same functions. That is, if

$$f_1^M(x_1, \ldots, x_n) = f_2^M(x_1, \ldots, x_n), \forall x_i \in \{0, 1\} \tag{58}$$

then

$$f_1^{CFB}(x_1, \ldots, x_n) = f_2^{CFB}(x_1, \ldots, x_n), \forall x_i \in [0, 1]. \tag{59}$$

This property is derived from assignment process of g. By equation (39), fuzzy measure values of basic sets are assigned from binary inputs of a fuzzy/C. Therefore, by equation (58),

$$g_1(B) = g_2(B), \ \forall B \in \mathfrak{B}. \tag{60}$$

From equation (40),

$$g_1(D) = g_2(D), \forall D \in \mathfrak{D}. \tag{61}$$

Hence,

$$g_1(A) = g_2(A), \ \forall A \in 2^{X^*} \tag{62}$$

and (C) $\int h^* dg_1 =$ (C) $\int h^* dg_2, \ \forall h^*(i) \in [0, 1]$.

On binary inputs, fuzzy/Cs satisfy the complementary law, i.e., $x_i \wedge \neg x_i = 0$ and $x_i \vee \neg x_i = 1$. Therefore, CFB functions satisfy the complementary law.

For example,

$$
\begin{aligned}
f_1(x_1, x_2) =&(0.2 \wedge x_1) \vee (0.3 \wedge x_2) \vee (x_1 \wedge x_2) \vee (x_2 \wedge \neg x_2) \\
f_2(x_1, x_2) =&(0.2 \wedge x_1) \vee (0.3 \wedge x_2) \vee (x_1 \wedge x_2).
\end{aligned}
$$

The two functions output the same value on binary inputs,

$$f_1(1,1) = f_2(1,1) = 1.0, \ldots, f_1(0,0) = f_2(0,0) = 0. \tag{63}$$

By equation (58), $f_1^{CFB}(x_1,x_2) = f_2^{CFB}(x_1,x_2), \forall x_1, x_2 \in [0,1]$. As is well-known, f^M does not satisfy the property. For example $x_1 = 0.2, x_2 = 0.5$, then $f_1^M(0.2,0.5) = 0.5$ and $f_2^M(0.2,0.5) = 0.3$.

4.7 Monotonicity of CFB Functions

CFB functions are monotone: $\forall x_i \in \{0,1\}, i = 1, \ldots, k-1, k+1, \ldots, n$,

$$f^M(x_1, \ldots, x_{k-1}, 1, x_{k+1}, \ldots, x_n) \geq f^M(x_1, \ldots, x_{k-1}, 0, x_{k+1}, \ldots, x_n), \tag{64}$$

implies, for any fixed $\bar{x}_i(\in [0,1], i = 1, \ldots, k-1, k+1, \ldots, n)$ and for any x_k, y_k where $1 \geq y_k \geq x_k \geq 0$,

$$f^{CFB}(\bar{x}_1, \ldots, \bar{x}_{k-1}, y_k, \bar{x}_{k+1}, \ldots, \bar{x}_n) \geq f^{CFB}(\bar{x}_1, \ldots, \bar{x}_{k-1}, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_n). \tag{65}$$

If the sign of inequality of equation (64) is reverse, the reverse sign of equation (64) is stood on.

For example, in equation (1), as $(f^M(1,1) = 0.8) \geq (f^M(0,1) = 0.8)$ and $(f^M(1,0) = 0.6) \geq (f^M(0,0) = 0.1)$, $f^{CFB}(x_1,x_2)$ is a monotone function of x_1 for any fixed x_2. Moreover, as $(f^M(1,1) = 0.8) \geq (f^M(1,0) = 0.6)$ and $(f^M(0,1) = 0.8) \geq (f^M(0,0) = 0.1)$, $f^{CFB}(x_1,x_2)$ is a monotone function of x_2 for any fixed x_1.

Fig. 5 is the graph of each calculation method. $f^M(0.7,x_2)$, max-min calculations, decreases near 0.5, but $f^{CFB}(0.7,x_2)$ increases in all $x_2 \in [0,1]$.

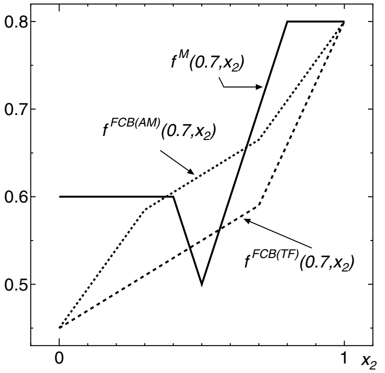


Fig. 5. Outputs of equation (1).

Another example is equation (29). In this example, $X^* = \{1^T, 1^F\}$,$g(\{1^T\}) = 1.0$, $g(\{1^F\}) = 0.6$,$h^*(1^T) = x_1$ and $h^*(1^F) = 1 - x_1$. As partition pattern of

$\{1^T, 1^F\}$ is unique, $g^{AM}(\{1^T, 1^F\}) = g^{TF}(\{1^T, 1^F\}) = g(\{1^T\}) + g(\{1^F\}) = 1.6$. Therefore, when $x_1 \geq 0.5$,

$$\begin{aligned} & f^{CFB(AM)}(x_1, x_2) = f^{CFB(TF)}(x_1, x_2) = \text{(C)} \int h^* dg \\ =&[h^*(1^T) - h^*(1^F)]g(\{1^T\}) + [h^*(1^F)g(\{1^T, 1^F\})] \\ =&[x_1 - (1 - x_1)] \times 1.0 + (1 - x_1) \times 1.6 \\ =&0.4 \times x_1 + 0.6, \end{aligned}$$

and when $x_1 < 0.5$,

$$\begin{aligned} & f^{CFB(AM)}(x_1, x_2) = f^{CFB(TF)}(x_1, x_2) = \text{(C)} \int h^* dg \\ =&[h^*(1^F) - h^*(1^T)]g(\{1^F\}) + [h^*(1^T)g(\{1^T, 1^F\})] \\ =&[(1 - x_1) - x_1] \times 0.6 + x_1 \times 1.6 \\ =&0.4 \times x_1 + 0.6. \end{aligned}$$

Hence, $f^{CFB(AM)}(x_1, x_2) = f^{CFB(TF)}(x_1, x_2) = 0.4 \times x_1 + 0.6$, which is the expected output of the Fig. 3.

4.8 Representations of Linear Functions

CFB functions can represent any linear functions $f^L : [0,1]^n \to [0,1]$.

Fuzzy measure values of basic sets are assigned by the method described in §4.2,

$$\begin{aligned} g(B) = f^L(x_1, \ldots, x_n) & \\ x_i = 1 \qquad & \text{if } \ i^T \in B \\ x_i = 0 \qquad & \text{if } \ i^F \in B, \\ \forall B \in \mathfrak{B}. & \end{aligned} \tag{66}$$

Other fuzzy measure values are assigned from equation (37),(38) and (40). In equation (40), by the additivity of f^L, $g(D)$ does not depend on GAV function, that is $g^{AM}(A) = g^{TF}(A) = g(A), \ \forall A \in 2^{X^*}$. Using g,

$$\text{(C)}h^* dg^{AM} = \text{(C)}h^* dg^{TF} = f^L(x_1, \ldots, x_n), \tag{67}$$

where $h^*(1^T) = x_i, h^*(1^F) = 1 - x_i, \forall i$.

Using g, it is possible to make fuzzy/C representation of any f^L,

$$f(x_1, \ldots, x_n) = \bigvee_{B \in \mathfrak{B}} [g(B) \wedge (\bigwedge_{i \in \{i | i^T \in B\}} x_i) \wedge (\bigwedge_{i \in \{i | i^F \in B\}} \neg x_i)]. \tag{68}$$

and

$$f^{CFB}(x_1, \ldots, x_n) = f^L(x_1, \ldots, x_n), \forall x_i \in [0,1]. \tag{69}$$

For example,

$$f^L(x_1, x_2) = 0.3 \times x_1 - 0.4 \times x_2 + 0.5 \tag{70}$$

is a linear function of $f^L : [0,1]^2 \to [0,1]$. Fuzzy measure values on basic sets given by equation (70) are,

$$g(\{1^T, 2^T\}) = 0.4, \ \ g(\{1^T, 2^F\}) = 0.8,$$
$$g(\{1^F, 2^T\}) = 0.1, \ \ g(\{1^F, 2^F\}) = 0.5.$$

From equation (40),

$$g(\{1^T, 1^F, 2^T\}) = 0.5, \ \ g(\{1^T, 1^F, 2^F\}) = 1.3,$$
$$g(\{1^T, 2^T, 2^F\}) = 1.2, \ \ g(\{1^F, 2^T, 2^F\}) = 0.6,$$

and $\{1^T, 1^F, 2^T, 2^F\}$ have two partition patterns, but the fuzzy measure values do not depend here on partition pattern, i.e.,

$$\begin{aligned} &g(\{1^T, 1^F, 2^T, 2^F\}) \\ =&g(\{1^T, 2^T\}) + g(\{1^F, 2^F\}) = g(\{1^T, 2^F\}) + g(\{1^F, 2^T\}) = 0.9. \end{aligned}$$

For example, $f^L(0.8, 0.4) = (\mathrm{C})h^* dg = 0.58$ where $h^*(1^T) = 0.8$, $h^*(1^F) = 0.2$, $h^*(2^T) = 0.4$ and $h^*(2^F) = 0.6$.

The fuzzy/C representation of equation (70) is,

$$f(x_1, x_2) = (0.4 \wedge x_1 \wedge x_2) \vee (0.8 \wedge x_1 \wedge \neg x_2) \vee (0.1 \wedge \neg x_1 \wedge x_2) \vee (0.5 \wedge \neg x_1 \wedge \neg x_2).$$

Then,

$$f^{CFB}(x_1, x_2) = f^L(x_1, x_2), \forall x_1, x_2 \in [0,1]. \tag{71}$$

CFB functions are the bridge between switching functions and linear functions, i.e., any linear function can be represented as a fuzzy/C formula and some fuzzy/C can be represented as a linear function.

4.9 Global Evaluation

Max-min calculation type fuzzy/Cs (f^M) satisfy the connectivity condition. For example, when

$$f_1(x_1, x_2) = x_1 \wedge x_2, \tag{72}$$
$$f_2(x_1, x_2) = x_1 \wedge \neg x_2, \tag{73}$$
$$f_3(x_1, x_2) = (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2), \tag{74}$$
$$f_4(x_3, x_4) = x_3 \vee x_4, \tag{75}$$
$$f_5(x_1, x_2) = x_1, \tag{76}$$

and

$$x_3^M = f_1^M(x_1, x_2), \tag{77}$$
$$x_4^M = f_2^M(x_1, x_2), \tag{78}$$

then,

$$f_3^M(x_1, x_2) = f_4^M(x_3^M, x_4^M). \tag{79}$$

However, CFB functions do not satisfy the connectivity condition.

$$x_3^{CFB} = f_1^{CFB}(x_1, x_2), \tag{80}$$

$$x_4^{CFB} = f_2^{CFB}(x_1, x_2), \tag{81}$$

do not imply

$$f_3^{CFB}(x_1, x_2) = f_4^{CFB}(x_3^{CFB}, x_4^{CFB}). \tag{82}$$

As CFB functions are based on global evaluations of the formulas, CFB functions are not based on each term calculation. Therefore, CFB functions can not heap up the formulas. CFB functions consider the global relations among logical variables. For example, f_3^{CFB} does not evaluate terms, $(x_1 \wedge x_2)$ and $(x_1 \wedge \neg x_2)$, but it uses global evaluation on binary inputs, $f_3^M(1, 1)$, $f_3^M(1, 0)$, $f_3^M(0, 1)$ and $f_3^M(0, 0)$. As fuzzy measure values on basic sets are assigned from binary inputs, CFB functions can satisfy the law of excluded middle ($x_i \vee \neg x_i = 1$). However, only by means of f_4, CFB functions can not understand the relation of $x_2 \vee \neg x_2$. To understand the law of the excluded middle, we have to evaluate the formula globally. Therefore, in a CFB function, we give up the connectivity and get the complementary law.

4.10 Constants

In fuzzy/C, max-min calculations, constants and variables are not distinguished For example, when

$$f_1(x_1, x_2) = x_2 \vee x_1, \;\; f_2(x_1) = 0.3 \vee x_1, \tag{83}$$

we have $f_1^M(x_1, 0.3) = f_2^M(x_1), \forall x_1 \in [0, 1]$. However, CFB functions do not satisfy this property: for instance, when $f_1^{CFB(TF)}(0.6, 0.3) = 0.3$, $f_1^{CFB(AM)}(0.6, 0.3) = 0.4$ and $f_2^{CFB(TF)}(0.6) = f_2^{CFB(AM)}(0.6) = 0.44$, then, $f_1^{CFB}(x_1, 0.3) = f_2^{CFB}(x_1)$, $\forall x_1 \in [0, 1]$ is not valid. Therefore, in CFB functions, constants are not fixed values of variables. We propose that constants imply weights of rules. This statement is discussed in §4.11.

In CFB functions, complementary law holds for variables, but not for constants. if $f_1(x_1) = x_1 \wedge \neg x_1$ and $f_2(x_1) = 0$ then $f_1^{CFB}(x_1) = f_2^{CFB}(x_1), \forall x_1 \in [0, 1]$. However, $f_3 = 0.8 \wedge 0.2$ and $f_4 = 0$ do not mean $f_3^{CFB} = f_4^{CFB}$, that is, $f_3^{CFB} = 0.2$.

4.11 Interpretation of CFB Functions

In §4.6, CFB functions satisfy the complementary laws. Using this property, we can easily find the disjunctive normal form f_{DN}, i.e.,

$$f_{DN}(x_1, \ldots, x_n) = \bigvee_{B \in \mathfrak{B}} [g(B) \wedge (\bigwedge_{i \in \{i | i^T \in B\}} x_i) \wedge (\bigwedge_{i \in \{i | i^F \in B\}} \neg x_i)]. \tag{84}$$

This transformation is valid in CFB functions,

$$f_{DN}^{CFB}(x_1,\ldots,x_n) = f^{CFB}(x_1,\ldots,x_n), \forall x_i \in [0,1], i = 1,\ldots,n, \tag{85}$$

but it is not valid in the case of max-min calculation f^M.

For example, the disjunctive normal form of equation (1) is,

$$\begin{aligned} f_{DN}(x_1,x_2) =&(0.8 \wedge x_1 \wedge x_2) \vee (0.6 \wedge x_1 \wedge \neg x_2) \\ &\vee (0.8 \wedge \neg x_1 \wedge x_2) \vee (0.1 \wedge \neg x_1 \wedge \neg x_2). \end{aligned} \tag{86}$$

This disjunctive normal form is very useful in understanding the fuzzy/C. In the disjunctive normal form, we can regard each term as a rule and its constant as a weight of the rule. The integration method of the rules is the Choquet integral.

For example, from equation (86), we can make rules shown in Table 3.

Table 3. Rule Table of Equation (1).

x_1	x_2	Ture Value
T	T	0.8
T	F	0.6
F	T	0.8
F	F	0.1

4.12 Independency of Input Sign

One of advantage of AM type CFB function is "independency of input sign". Let $f_1(x_1,\ldots,x_n)$ be a fuzzy/C and $f_2(x_1,\ldots,x_2)$ be the function that has x_k replaced with $\neg x_k$. "Independency of input sign" is defined as,

$$f_1^{CFB}(x_1,\ldots,x_n) = f_2^{CFB}(x_1,\ldots,x_{k-1},1-x_k,x_{k+1},\ldots,x_n), \tag{87}$$

$\forall x_i \in [0,1]$. CFB(AM) satisfies this property, i.e.,

$$f_1^{CFB(AM)}(x_1,\ldots,x_n) = f_2^{CFB(AM)}(x_1,\ldots,x_{k-1},1-x_k,x_{k+1},\ldots,x_n), \tag{88}$$

$\forall x_i \in [0,1]$. For example,

$$f_1(x_1,x_2) = (0.6 \wedge x_1 \wedge \neg x_2) \vee (0.8 \wedge x_2) \vee 0.1, \tag{89}$$

$$f_2(x_1,x_2) = (0.6 \wedge x_1 \wedge x_2) \vee (0.8 \wedge \neg x_2) \vee 0.1, \tag{90}$$

where equation (90) is the fuzzy/C in which x_2 is replaced with $\neg x_2$. Therefore,

$$f_1^{CFB(AM)}(x_1,x_2) = f_2^{CFB(AM)}(x_1,1-x_2), \forall x_1,x_2 \in [0,1]. \tag{91}$$

However, TF type does not satisfy this property, for example,

$$f_1^{TF}(0.7,0.4) = 0.53, \;\; f_2^{TF}(0.7,0.6) = 0.68. \tag{92}$$

5 Logical Choquet Integral

5.1 Definition

A CFB function is characterized by the a fuzzy measure g; g have $2^{2\times n}$ parameters, but g is characterized by fuzzy measure values basic sets only. The number of basic sets is 2^n, which is the same as the number of parameters of μ. Basic sets include either i^T or i^F for each $i \in X$. Therefore, a basic set corresponds to a subset of X. For example, $\{1^T, 2^F, 3^T\}$ $(n = 3)$ corresponds to $\{1, 3\}$. Therefore, we regard $\mathfrak{B}$ as a domain of a fuzzy measure and define fuzzy measure μ^+,

$$\mu^+ : 2^X \to [0, 1] \tag{93}$$

$$\mu^+(A) = g(B),\ \ B = \{i^T \mid i \in A\} \cup \{i^F \mid i \notin A\}, \tag{94}$$

$\forall A \in 2^X$. μ^+ does not satisfy the monotonicity of fuzzy measure and the zero constraint of fuzzy measure.

CFB functions are the functions which map $[0, 1]^n$ to $[0, 1]$ and are characterized by μ^+, so we define the new fuzzy integral and we call it the logical Choquet integral (LC integral, for short),

$$(\mathrm{LC})\int h d\mu^+ \equiv (\mathrm{C})\int h^* dg \tag{95}$$

where

$$g(B) = \mu^+(A),\ \ A = \{i \mid i^T \in B\},\ \ \forall B \in \mathfrak{B},$$

$$g(D) = \underset{\{(B_1,B_2)\mid B_1\cup B_2=D, B_1,B_2\in\mathfrak{B}\}}{\mathrm{GAV}}[g(B_1) + g(B_2)],\ \ \forall D \in \mathfrak{D},$$

$$g(A) = 0,\ \ \forall A \in [2^{X^*} \setminus (\mathfrak{B} \cup \mathfrak{D})],$$

$$\begin{aligned} h^*(i^T) &= \quad h(i), i = 1, \ldots, n, \\ h^*(i^F) &= 1 - h(i), i = 1, \ldots, n. \end{aligned}$$

An LC integral satisfies all properties of CFB functions. Whenever we specify the GAV function, we add superscript of GAV function, that is, $(\mathrm{LC}^{\mathrm{AM}})\int h d\mu^+ \equiv (\mathrm{LC})\int h^* dg^{AM}$ and $(\mathrm{LC}^{\mathrm{TF}})\int h d\mu^+ \equiv (\mathrm{C})\int h^* dg^{TF}$.

μ^+ is also assigned from fuzzy/C's binary inputs, i.e.,

$$\begin{aligned} \mu^+(A) = f^M(x_1, \ldots, x_n) \quad , \quad \forall A \in 2^X \\ x_i = 1 \ \ \text{if} \ \ i \in A \\ x_i = 0 \ \ \text{if} \ \ i \notin A. \end{aligned} \tag{96}$$

In equation (1), $n = 2$ and $X = \{1, 2\}$. From equation (52) $\sim$ (55),

$$\mu^+(\emptyset) = f^M(0,0) = 0.1, \tag{97}$$
$$\mu^+(\{1\}) = f^M(1,0) = 0.6, \tag{98}$$
$$\mu^+(\{2\}) = f^M(0,1) = 0.8, \tag{99}$$
$$\mu^+(\{1,2\}) = f^M(1,1) = 0.8. \tag{100}$$

Obviously, μ^+ is the same as μ defined in equation (16).

5.2 The Extended Choquet Integral

In §3.2, we define the extend Sugeno integral which avoids the condition that $\mu(\emptyset) = 0$. Therefore, we define the extended Choquet integral which avoids $\mu(\emptyset) = 0$, i.e., the Choquet integral of μ^+,

$$(\mathrm{EC})\int h d\mu^+ \equiv \int_0^1 \mu^+(\{x \mid h(x) > r\})dr. \tag{101}$$

In case $n = 2$, equation (97)$\sim$(100) and $h(1) = 0.6, h(2) = 0.3$,

$$\begin{aligned}&(\mathrm{EC})\int h d\mu^+\\ =&(1-0.6)\mu^+(\emptyset) + (0.6-0.3)\mu^+(\{1\}) + 0.3\mu^+(\{1,2\})\\ =&0.46.\end{aligned}$$

5.3 LC$^{\mathbf{TF}}$ Integrals and Extended Choquet Integrals

LC$^{\mathrm{TF}}$ integrals are extended Choquet integrals,

$$(\mathrm{LC}^{\mathrm{TF}})\int h d\mu^+ = (\mathrm{EC})\int h d\mu^+, \forall h \text{ and } \mu^+. \tag{102}$$

Therefore, TF type CFB functions are calculated by the extended Choquet integral, that is, by a very easy calculation method:

$$f^{CFB(TF)}(x_1, \ldots, x_n) = (\mathrm{EC})\int h\mu^+. \tag{103}$$

Of course, if $f^M(x_1, \ldots, x_n) = 0$, then

$$f^{CFB(TF)}(x_1, \ldots, x_n) = (\mathrm{C})\int h\mu^+. \tag{104}$$

6 Comparison with Probability Based Logic

Multi-valued switching functions which are based on probability also satisfy the complementary law. For example, Tsukimoto[5] define "T-Logic" as,

$$f \wedge g \to f \times g, \tag{105}$$

$$f \vee g \to f + g - f \times g, \tag{106}$$

$$\neg f \to 1 - f. \tag{107}$$

We add superscript ^{P} to the "T-Logic" calculations.

This formulation conflicts with our intuition. For example,

$$f_1(x_1, x_2) = x_1 \wedge x_2 \tag{108}$$

means that "if both x_1 and x_2 are true, then output is true" and also means "if both x_1 and x_2 are half true, then output is half true". As $f_1^P(x_1, x_2) = x_1 \times x_2$, $f_1^P(0.5, 0.5) = 0.25$, that is not half trueness, because T-logic base on the probability. However, $f_1^M(0.5, 0.5) = f_1^{CBF}(0.5, 0.5) = 0.5$.

Equation (108) means non-compensated outputs, that is, if $x_1 > x_2$, increase in x_1 does not cause increase in f_1. f^M and f^{CFB} satisfy non-compensated outputs but f^P does not, i.e., it satisfies compensated outputs. If compensated outputs are necessary, we should consider a compensated formula such as,

$$f_2(x_1, x_2) = (x_1 \wedge x_2) \wedge (0.2 \wedge x_1) \vee (0.1 \wedge x_2), \tag{109}$$

which means that "an important rule is $x_1 \wedge x_2$, but x_1 or x_2 alone also stand on with small weights". f_2^{CFB} is a compensated function and a strictly increasing function for x_1 and x_2.

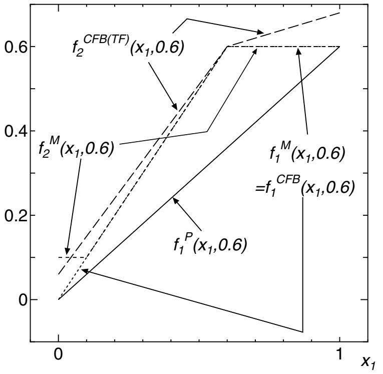


Fig. 6. Comparison with Probability Based Logic.

Fig. 6 is the graph of $x_2 = 0.6$. f_1^M and f_1^{CFB} are non-compensated with $x_1 \geq 0.6 = x_2$, that is $f_1^M(x_1, 0.6) = f_1^{CFB}(x_1, 0.6) = 0.6, \forall x_1 \in [0.6, 1]$. f^P

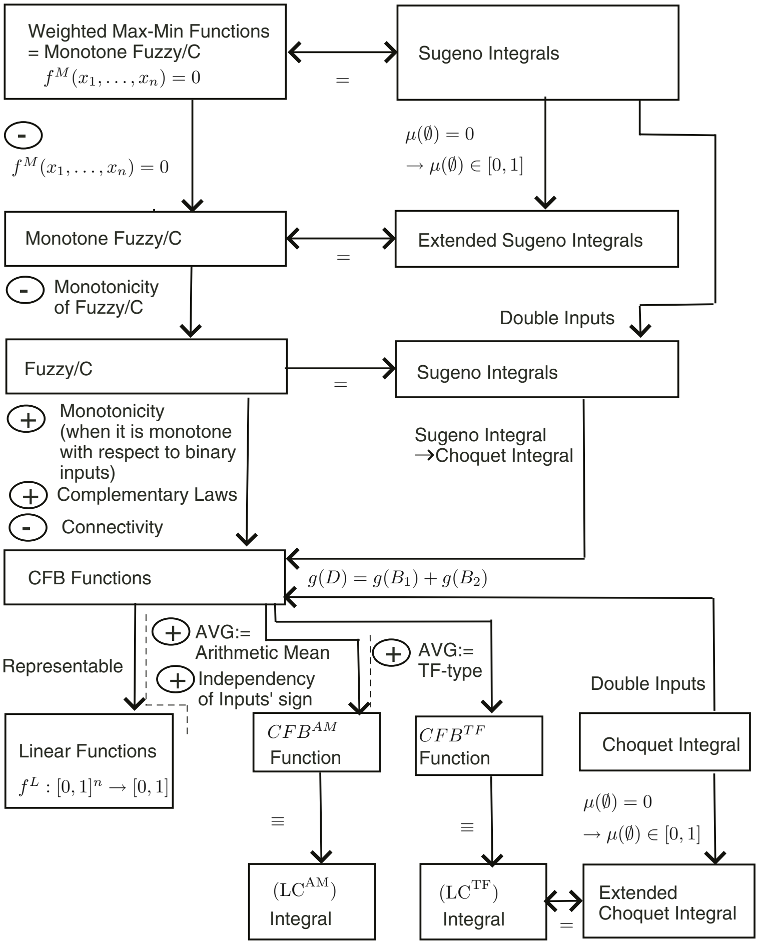

Weighted Max-Min Functions = Monotone Fuzzy/C
$f^M(x_1,\ldots,x_n)=0$
Sugeno Integrals
=
-
$f^M(x_1,\ldots,x_n)=0$
$\mu(\emptyset)=0$
$\to \mu(\emptyset)\in[0,1]$
Monotone Fuzzy/C
Extended Sugeno Integrals
=
- Monotonicity of Fuzzy/C
Double Inputs
Fuzzy/C
Sugeno Integrals
=
+ Monotonicity (when it is monotone with respect to binary inputs)
+ Complementary Laws
- Connectivity
Sugeno Integral →Choquet Integral
CFB Functions
$g(D)=g(B_1)+g(B_2)$
+ AVG:= Arithmetic Mean
+ Independency of Inputs' sign
+ AVG:= TF-type
Representable
Double Inputs
Linear Functions
$f^L:[0,1]^n\to[0,1]$
CFB^{AM} Function
CFB^{TF} Function
Choquet Integral
≡
≡
$\mu(\emptyset)=0$
$\to \mu(\emptyset)\in[0,1]$
$(\mathrm{LC}^{\mathrm{AM}})$ Integral
$(\mathrm{LC}^{\mathrm{TF}})$ Integral
=
Extended Choquet Integral

Fig. 7. The Relations of Fuzzy/Cs and Fuzzy Integrals.

increases in all x_1's domain without regard to $x_1 > x_2$. f_2^{CFB} increases across the whole domain of x_1. When $x_1 \leq x_2$, the important rule is valid and the increase rate is high; when $x_1 > x_2$, only small weight rule is observed and the increase rate is low.

From above reasons, CFB functions have superior properties.

7 Conclusion

In this paper, we define many fuzzy/C's calculation methods using fuzzy integrals. Max-min calculation methods do not satisfy the complementarity law. Also max-min calculations f^M do not satisfy monotonicity under monotonicity on binary inputs.

CFB functions calculate using fuzzy/C's binary inputs' values and interpolate those values. Therefore, CFB functions satisfy the many properties of (2-valued) switching functions. Hence, in various analysis which uses the fuzzy theory, whenever fuzziness is an extension of 2-values, we should use CFB functions.

Fig. 7 is the conclusion of this paper. As fuzzy/Cs are represented by Sugeno integrals, we can consider properties of fuzzy/C via tools of fuzzy measures and Sugeno integrals.

CFB(AM) functions have many good properties, but the computational effort of $g(D), D \in \mathfrak{D}$ is large. Because, if n is large, partition patterns are numerous, so arithmetic means can not be calculated easily. However, CFB(TF) functions are calculated by extended Choquet integrals, resulting in a small computational effort. Therefore, we should use CFB(AM) in case of strict analysis and use CFB(TF) in case of large model.

References

1. J.L. Marichal, On Sugeno integral as an aggregation function. *Fuzzy Sets and Systems*, **114**, 2000, pp.347–365.
2. E. Takahagi and T. Araki. On fuzzy Integral representation in fuzzy switching functions with constants. in: *Proceedings of Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications (VJFUZZY'98)*, Hanoi, 1998, pp.240–245.
3. E. Takahagi. On fuzzy integral representation in fuzzy switching functions, fuzzy rules and fuzzy control rules. in: *Proceedings of the Eighth International Fuzzy Systems Association World Congress(IFSA'99)*, **1**, Taipei, 1999, pp.289–293.
4. E.Takahagi. Choquet integral type fuzzy switching functions: Adjustment of fuzzy measures and complementary law. in: *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, CD-ROM, IEEE, Vancouver, 2001.
5. H. Tsukimoto. On continuously valued logical functions satisfying all axioms of classical logic. *The Transactions of the Institute of Electronics, Information and Communication Engineers D-I*, **J77-D-I: 3**, 1994, pp.247–252 (in Japanese).
6. T. Araki and M. Mukaidono. On the fuzzy switching functions with constants, *The Transactions of the Institute of Electronics, Information and Communication Engineers D-I*, **J81-D-I: 9**, 1998, pp.1037-1047 (in Japanese).

First Steps Towards Computably-Infinite Information Systems

Peter Apostoli[1], Akira Kanda[2], and Lech Polkowski[3,4]

[1] The University of Pretoria, Pretoria, South Africa
[2] The University of Toronto, Ontario, M5S 1A2, Canada
[3] Polish Japanese Institute of Information Technology, Warsaw, Poland
[4] The University of Warmia and Mazury, Olsztyn, Poland

Abstract. In order to characterize the metric of exact subsets of infinite information systems, [51] studied the asymptotic behaviour of ω–chains of graded indiscernibility relations. The SFP object underlying the universe of exact sets presented in [2] provides a concrete example of an infinite graded information system. By controlling the asymptotic behaviour of ω–Sequences of Finite Projections, the theory of graded chains of indiscernibility relations articulates the fine structure of SFP objects, providing a metric over exact sets.

Keywords: domain theory, effective domain theory, exact sets, Frege sets, graded indiscernibility, infinite information systems, Myhill-Sheperdson Theorem, modal logic, non–standard analysis, numeration theory, rough sets, SFP objects.

1 Infinite Information Systems

Though distinct fields of computer science, Rough Set Theory (RST) [44] and Domain Theory (DT) [58] both study the approximation of concepts in infinite information systems. The former studies the topological approximation of subsets by upper and lower bounding subsets, while the latter studies the order–theoretic approximation of functions under an information ordering. However their techniques diverge, these fields are intricately related when attention is focused on infinite information systems arising in connection with the inverse limits of Sequences of Finite Projections (SFP) [47].

Given a set of points U, RST studies the field of subsets of U under upper and lower approximation operations, typically those of the quasi–discrete topology induced by an equivalence relation E on U [44]. In the finite case, the field of exact subsets forms a compact metric Boolean algebra. In order to extend this characterization to the exact subsets of infinite information systems [51] studied the asymptotic behaviour of approximations by descending chains of equivalence relations $\{E_i\}$ over U such that $E_{i+1} \subseteq E_i$, called "graded chains of indiscernibility relations". The limit E_∞ of this chain is an equivalence relation called *the indiscernibility relation graded by* $\{E_i\}$. The collection of all E_i-granules forms a base for a "graded" topology in [51]. We shall show that this graded topology is identical to the Pawlak topology induced by E_∞.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 151–188, 2004.

By integrating the "distance" between two elements of U as the stage at which they are discerned in the chain $\{E_i\}$, [51] obtains a metric distance – which we call a *graded metric* – between subsets of U which are closed in the graded topology. As Pawlak's topology is quasi–discrete, we conclude that the "closed" subsets are precisely the exact subsets of U whence the graded metric obtains over the exact subsets of U. As an example of an infinite information system based upon graded indiscernibility, we use the theory of SFP objects to construct a universe of exact sets by placing an approximation space (U, E) in a type–lowering retraction with 2^U. Accordingly, this universe of exact sets is accompanied by a graded metric.

By dualizing Polkowski's construction, the concept of an *inversely graded* indiscernibility relation and metric is obtained. A. Robinson's [54] ultra product construction of non–standard real numbers provides a viable example of this. This conclusion is consistent with the analogy developed in [4] that the universe of exact sets is to standard set theory as Robinson's non–standard analysis is to standard analysis. We hope that this development suggests a foundational role for the study of infinite information systems in rough set theory and analysis.

2 Rough Set Theory

Let $U \neq \emptyset$ and $E \subseteq U \times U$ be an equivalence relation. In rough set theory [44], E is typically interpreted as a relation of *indiscernibility* with respect to some prior family of concepts (subsets of U). The pair (U, E) is called an *approximation space*. For each point $u \in U$, let $[u]_E$ denote the equivalence class of u under E. E-classes $[u]_E$ are called (E-) *granules* or *elementary* subsets. Let $A \subseteq U$;

$$\begin{aligned} Int(A) &=_{df} \{x \in U \mid (\forall y \in U)[xE\,y \rightarrow y \in A]\} = \cup\{[u]_E \mid [u]_E \subseteq X\}, \\ Cl(A) &=_{df} \{x \in U \mid (\exists y \in U)[xE\,y \wedge y \in A]\} = \cup\{[u]_E \mid [u]_E \cap X \neq \emptyset\}, \end{aligned}$$

are called the *lower* and *upper approximations* of A, respectively. $Cl(A)$ is a relational closure under E of the subset A. E-closed subsets of U are called (E-) *complete (or exact)*. It is natural to regard the exact subsets of U as the "parts" of U and elementary subsets as the "atomic parts" of U [1]. $\mathcal{C}(E)$ denotes the family of E–exact subsets of U and, for $A \subseteq U$, $\mathcal{C}(A)$ the family of E–exact subsets of A. The family,

$$\mathcal{B}_E = \{[x]_E : x \in U\},$$

is the partition of U induced by the equivalence relation E and it forms a base for a quasi–discrete topology Π_E over U. It is clear that the open sets in this topology are precisely the exact sets.

[1] RST marks the re–emergence in the Warsaw School of Mathematics of mereological concepts such as "part" and "whole" that were originally introduced into set theory by Leśniewski [36]; see e.g., rough mereology [48–50]. For an example of the mereological conception of set in the recent philosophical literature – motivated independently of RST – see [37].

Given an approximation space (U, E) and $A, B \subseteq U$, [44] defines A and B to be *roughly equal*, symbolically, $A \sim_E B$, iff $Int(A) = Int(B)$ and $Cl(A) = Cl(B)$. Hence, $A \sim_E B$ iff A and B are equivalent with respect to the partition topology $(U, \mathcal{C}(E))$ induced by (U, E). Rough equality classes $[A]_{\sim_E}$ are called *topological rough sets* [63]. In the context of (U, E), $[A]_{\sim_E}$ may be identified with the interval of subsets lying between $Int(A)$ and $Cl(A)$, in which A is approximated from "below" by its interior and from "above" by its closure.

3 Proximal Frege Structures

The comprehension principle of naive set theory is based upon Frege's [21, 22] ill–fated [23–25] idea that for every concept X there is an object, $\widehat{X}$, called *the extension* of X, which comprises precisely the objects falling under X. $\widehat{X}$ is, intuitively, the set of objects u such that u falls under X, i.e., $\{u : X(u)\}$. Frege thus posited the existence of a "type–lowering correspondence", terminology established in [5] for functions $f : 2^U \to U$, holding between the concept universe 2^U and the universe of objects U. Frege attempted to govern his introduction of extensions by adopting the principle that $\widehat{X}$ and $\widehat{Y}$ are strictly identical just in case precisely the same objects fall under the concepts X and Y. That this "Basic Law" contradicts Cantor's Theorem[2] in requiring f to be injective and leads to Russell's paradox regarding "the class of all classes which do not belong to themselves" [23, 55, 25], is now well appreciated. [15] initiated the proof theoretic study of Frege's extension function in consistent theories extending second order logic. Models of these theories are obtained by placing a domain of discourse U in a type–lowering correspondence with its power set 2^U and are called "Frege structures" after terminology established in [1]. The rediscovery of Frege's Theorem[3] [43, 64, 9] in type–lowering extensions of second order logic re–focused interest on Frege structures. [3, 2, 6] presented Frege structures as *retraction pairs* of type–lowering and raising maps, i.e., pairs (f, g) of functions $f : 2^U \to U$, $g : U \to 2^U$ such that $f(g(u)) = u$. f retracts 2^U onto U (determining sethood) and g is the adjoining section (determining elementhood). The Frege structure (U, f, g) is a model of abstract set theory in which Cantor's "consistent multiplicities" [13, p. 443.] can be identified with precisely those subsets of U which are elements of the section $g[U]$ of the retraction, whence the fundamental question [6] arises: which subsets are *these*?

[2] According to Cantor's Theorem [12], the power set 2^U is of strictly larger cardinality than U.

[3] In the *Grundlagen der Arithmetik* [21], Frege sketched a derivation of Peano's postulates for the arithmetic of natural numbers from Cantor's Principle of equality for the cardinal numbers [11], that sets have the same cardinal number just in case they are equipollent. Called "Hume's Principle" (HP) by Boolos [9], this principle is presented in *Grundlagen* §63 as the statement that, for any concepts F and G: the number of Fs = the number of Gs iff F is equipollent with G. The derivation of the infinity of the natural number series in second order logic + HP − a theory Boolos calls "Frege's Arithmetic" − is now called "Frege's Theorem" [9].

One concrete answer is provided along the lines of Rough Set Theory. Accordingly, one may augment the retraction pair (f, g) with an equivalence relation E on U and characterize the section as comprised of precisely the E–closed subsets of U [2]. Comprehension is governed by the method of upper and lower approximations in the resulting universe (U, E, f, g) of *abstract* sets, called [2] a "proximal" Frege structure. The leading idea is that logical space has an atomic structure and this granularity is the form of set theoretic comprehension.

Let (U, E) be an approximation space and $\ulcorner \cdot \urcorner : 2^U \to U$, $\llcorner \cdot \lrcorner : U \to 2^U$ be functions. Assume further,

1. $\ulcorner \cdot \urcorner$ is a retraction, i.e., $\ulcorner \llcorner u \lrcorner \urcorner = u$ (i.e., $\ulcorner \cdot \urcorner \circ \llcorner \cdot \lrcorner = 1_U$), thus $\llcorner \cdot \lrcorner$ is the adjoining section.
2. The operator $\llcorner \cdot \lrcorner \circ \ulcorner \cdot \urcorner$ is a closure (under E). This is that for every $X \subseteq U$, $\llcorner \ulcorner X \urcorner \lrcorner$ is E–closed and
$$X \subseteq \llcorner \ulcorner X \urcorner \lrcorner.$$
So,
$$\llcorner \ulcorner X \urcorner \lrcorner = Cl(X).$$
3. The E–closed subsets of U are precisely the $X \subseteq U$ for which $\llcorner \ulcorner X \urcorner \lrcorner = X$. They are fixed–points of the operator $\llcorner \cdot \lrcorner \circ \ulcorner \cdot \urcorner$.

Then $(U, E, \ulcorner \cdot \urcorner, \llcorner \cdot \lrcorner)$ is called a *proximal Frege structure (PFS)*. Elements of U are called *Frege sets*. The indiscernibility relation E of a PFS is usually denoted $\equiv$. The fundamental idea behind this rather curious mathematical structure is to use the approximation space [44] to tame the reflexive universe U of sets,

$$U \lhd 2^U,$$

where "$U \lhd 2^U$" indicates that $\ulcorner \cdot \urcorner$ retracts the power set of U onto U.

It follows that for every $u \in U$,

$$\llcorner u \lrcorner = \llcorner \ulcorner \llcorner u \lrcorner \urcorner \lrcorner,$$

and hence, $\llcorner u \lrcorner$ is $\equiv$–exact. Suppose X is $\equiv$–closed, then $\llcorner \ulcorner X \urcorner \lrcorner = X$. Thus,

$$X \in \llcorner U \lrcorner = Image(\llcorner \cdot \lrcorner).$$

So, the set of $\equiv$–closed subsets of U is precisely the image of $\llcorner \cdot \lrcorner$. In algebraic terms it is the kernel of the retraction mapping.

Further, we have the isomorphism $\mathcal{C}(\equiv) \approx U$ given by,

$$i : \mathcal{C}(\equiv) \to U : X \mapsto \ulcorner X \urcorner,\ j : U \to \mathcal{C}(\equiv) : u \mapsto \llcorner u \lrcorner.$$

All of this may be summarized by the equation,

$$\mathcal{C}(\equiv) \approx U \lhd 2^U,$$

where $\mathcal{C}(\equiv)$ is the family of $\equiv$-closed subsets of U.

Let $\mathcal{F} = (U, \equiv, \ulcorner \cdot \urcorner, \llcorner \cdot \lrcorner)$ be a proximal Frege structure. Writing "$u_1 \in_{\mathcal{F}} u_2$" for "$u_1 \in \llcorner u_2 \lrcorner$", U interpreted as a universe of abstract (or type–free)sets; $\llcorner \cdot \lrcorner$ supports the relation of set membership holding between type–free sets (elements of U). Note that as $\llcorner u_2 \lrcorner$ is $\equiv$–closed, whenever $u_1 \equiv v$,

$$u_1 \in \llcorner u_2 \lrcorner \leftrightarrow v \in \llcorner u_2 \lrcorner.$$

$\mathcal{F}$ thus validates the Principle of Naive Comprehension (PNC),

$$u \in_{\mathcal{F}} \ulcorner X \urcorner \Leftrightarrow X(u),$$

for exact ($\equiv$–closed) subsets X of U. When X fails to be elementary, the equivalence also fails and is replaced by a pair of approximating conditionals,

$$\begin{array}{l} (1)\ u \in_{\mathcal{F}} \ulcorner Int(X) \urcorner \Rightarrow X(u); \\ (2)\ X(u) \Rightarrow u \in_{\mathcal{F}} \ulcorner Cl(X) \urcorner. \end{array}$$

Note here we now use applicative grammar "$X(u)$" ("u falls under X) to indicate that u is an element of $X \subseteq U$. Let "$\{x : X(x)\}$" denote $\ulcorner X \urcorner$. While "$\{u \in U \mid X(u)\}$" denotes a *subset* of U, the expression "$\{x : X(x)\}$ denotes an *element* of U. We distinguish the indiscernibility class $[u]_\equiv$ of a Frege set u from Frege set $\ulcorner [u]_\equiv \urcorner$ that represents $[u]_\equiv$; the latter is denoted $\overline{\{u\}}$, and is called the *proximal singleton* of u. Let $x, y \in U$. Then, both

$$\begin{array}{l} (a)\ (\forall u)(u \in_{\mathcal{F}} x \leftrightarrow u \in_{\mathcal{F}} y) \leftrightarrow x = y; \\ (b)\ (\forall u)(x \in_{\mathcal{F}} u \leftrightarrow y \in_{\mathcal{F}} u) \leftrightarrow x \equiv y. \end{array}$$

Theorem 1. ***(Cocchiarella 1972)*** *There are $x, y \in U$ such that $x \equiv y$ but $x \neq y$.*

Proof. As assuming the failure of the theorem (i.e., assuming the identity of indiscernibles), [15, 16] derives Russell's contradiction directly, the theorem follows.

Cocchiarella's theorem ensures that whenever we have a PFS $\mathcal{F} = (U, \equiv, \ulcorner \cdot \urcorner, \llcorner \cdot \lrcorner)$, the indiscernibility relation $\equiv$ is coarser than the strict identity.

Since elements of U represent elementary subsets of U, the complete Boolean algebra,

$$(\mathcal{C}(\equiv), \cup, \cap, -, \emptyset, U),$$

is isomorphic to $(U, \cup_U, \cap_U, -_U, \ulcorner \emptyset \urcorner, \ulcorner U \urcorner)$ under the restriction $\ulcorner \cdot \urcorner \restriction \mathcal{C}(\equiv)$ of the type–lowering retraction to elementary subsets of U. Here, $\cup_U, \cap_U, -_U$ denote the definitions of union, intersection and complementation natural to type–free sets, e.g.,

$$u_1 \cup_U u_2 =_{df} \ulcorner \llcorner u_1 \lrcorner \cup \llcorner u_1 \lrcorner \urcorner$$

etc., (in most contexts the subscripted "U" is suppressed). We define "$u_1 \subseteq u_2$" to be "$\llcorner u_1 \lrcorner \subseteq \llcorner u_2 \lrcorner$", i.e., *inclusion* of Frege sets is the partial ordering naturally associated with the Boolean algebra of Frege sets.

3.1 Counterparts and Inner and Outer Penumbra

Since unions of $\mathcal{F}$-sets are exact,

$$\{x \in U \mid (\exists\, g \in U)[f \equiv g \wedge x \in_{\mathcal{F}} g]\},$$

is an exact subset of U. Let $a \in U$; define the *outer penumbra* of a, symbolically, $\Diamond a$, to be the $\mathcal{F}$-set $\cup[a]_{\equiv}$; dually, define the *inner penumbra*, $\Box a$, to be the $\mathcal{F}$-set $\cap[a]_{\equiv}$. These operations, called the *penumbral modalities*, are interpreted using David Lewis' counterpart semantics for modal logic [37]. Your "counterpart" (in a given world) is a person more qualitatively similar to you than any other object (in that world). You are necessarily (possibly) P iff all (some) of your counterparts are P. Similarly, if a and b are indiscernible $\mathcal{F}$-sets, a is more similar to b than any *other* (i.e., discernible!) $\mathcal{F}$-set. Thus we call a $\mathcal{F}$-set b a *counterpart* of a $\mathcal{F}$-set a whenever $a \equiv b$. Then $\Box a$ ($\Diamond a$) represents the set of $\mathcal{F}$–sets that belong to all (some) of a's counterparts. In this sense, we can say that a $\mathcal{F}$-set x *necessarily* (*possibly*) belongs to a just in case x belongs to $\Box a$ ($\Diamond a$).

Define $a\langle x\rangle =_{df} x \in_{\mathcal{F}} \Diamond a$; thus,

$$a\langle x\rangle \Leftrightarrow x \in_{\mathcal{F}} \Diamond a \Leftrightarrow x \in \llcorner \Diamond a \lrcorner.$$

This is the *outer membership relation*, also written $x \in_{\Diamond} a$. Thus, for all $a, x \in U$,

$$x \in_{\mathcal{F}} a \rightarrow x \in_{\Diamond} a.$$

It is equally clear that the converse of this does not hold. Dually, define $a[x] =_{df} x \in_{\mathcal{F}} \Box a$; thus,

$$a[x] \Leftrightarrow x \in_{\mathcal{F}} \Box a \Leftrightarrow x \in \llcorner \Box a \lrcorner.$$

This is the inner *membership relation*, also written $x \in_{\Box} a$.

The principle that disjoint sets are discernible is a triviality of classical set theory. However, it is independent of the axioms for PFS given thus far. *The Discernibility of the Disjoint* (DoD) [2, 3] is the principle that disjoint $\mathcal{F}$-sets be discernible.

We say that a PFS $\mathcal{F}$ is a *plenum* iff it satisfies the following condition of plenitude: For all $a, b \in U$, $\Box a \equiv a \equiv \Diamond a$, whence $\Box a \equiv \Diamond a$. Further, suppose $a \subseteq b$ and $a \equiv b$. Then, for all $c \in U$, $a \subseteq c \subseteq b \Rightarrow c \equiv b$. Plenitude is a consistent extension of the notion of a PFS which tells us that $\mathcal{F}$ inclusion is well–behaved with respect to indiscernibility. It is a nontrivial generalization, via "blurring", of a truism of classical set theory that if a is a subset of b and a is indiscernible from b, then any c which is in between a and b as a subset is indiscernible from b. If $\mathcal{F}$ is a plenum, then,

Lemma 1. *For all $a \in U$ and $X \subseteq U$ such that $x \equiv a$ $(x \in X)$, we have $\cup X$, $\cap X \equiv a$.*

Proof. $\Box a \subseteq \cup X, \cap X \subseteq \Diamond a$, so the desired result follows from Plenitude by the fact that $\Box a \equiv \Diamond a$.

Corollary 1. $([a]_{\equiv}, \cap_U, \cup_U, \Box a, \Diamond a)$ *is a complete lattice with least (greatest) element* $\Box a$ $(\Diamond a)$.

The canonical PFS M_{max} to whose domain theoretic construction we now turn is a concrete example of a plenum which satisfies DoD.

4 The Canonical PFS M_{max}

4.1 Basic Domain Theory

Cpo's and Continuous Functions. We assume the standard terminology and notation for partially ordered sets (posets). When a poset X has the smallest element, we denote it by $\bot_X$. We may drop the subscript X and write just $\bot$, as long as doing so does not cause confusion.

A poset X with smallest element such that every directed subset S of it has a least upper bound $\bigvee S$ (called a *directed limit of* S) in X is called a *complete partial order (cpo).* A singleton clearly is a cpo and we call it a *trivial cpo.* Let A and B be posets. A function $f : A \to B$ is *monotonic* (or *monotone*) if it is order preserving. Moreover, it is *continuous* if for each directed subset S of A,

$$f(\bigvee S) = \bigvee f(S) = \bigvee \{f(x) \mid x \in S\}.$$

It is clear that all continuous functions are monotonic. So, continuous functions are precisely those which preserve directed limits.

The collection of cpo's and that of monotonic functions form a category which we denote by $\mathcal{MCPO}$, abbreviated by $\mathcal{M}$. The collection of cpo's and that of continuous functions form a category which is denoted by $\mathcal{CCPO}$, abbreviated $\mathcal{C}$. $\mathcal{CCPO}$ is a full subcategory of $\mathcal{MCPO}$.

Retraction and Embedding. For cpo's A and B, we call a pair $(s : A \to B,\ r : B \to A)$ of monotonic (continuous) functions an *m(c)-retraction pair* (from B *to* A) iff

$$r \cdot s = 1_A;$$

s is called the *m(c)-section* and r is called the *m(c)-retraction.* If there is an m–retraction pair from A to B, we say A is an *m–retract of* B, in symbols,

$$A \lhd_{\mathcal{M}} B.$$

The concept of *c–retract*,

$$A \lhd_{\mathcal{C}} B,$$

can be defined similarly. If the retraction pair is a pair of morphisms in a subcategory $\mathcal{S}$ of $\mathcal{MCPO}$, we may write

$$A \lhd_{\mathcal{S}} B.$$

For cpo's A and B, let a pair $(f : A \to B, g : B \to A)$ of monotonic (continuous) functions be called an *m(c)–embedding pair (from A to B)* if

$$f \cdot g \leq 1_B, \quad g \cdot f = 1_A.$$

f is called the *m(c)–embedding* and g is called the *m(c)–projection*. Sometimes, embedding pairs are called *m(c)–projection pairs*. It can easily be verified that an m(c)–embedding uniquely determines the corresponding projection and vice versa. So, we may write f^R for g when (f, g) is an $m(c)$–embedding pair. If there is an $m(c)$–embedding $f : A \to B$, we write

$$A \curvearrowright B.$$

If the embedding pair is a pair of morphisms in a subcategory $\mathcal{S}$ of $\mathcal{MCPO}$, we may write

$$A \curvearrowright_{\mathcal{S}} B.$$

If it does not cause confusion, we will drop the subscript $m(c)$ in the foregoing.

It is clear that if $(f : A \to B, g : B \to A)$ is an m–embedding pair then (g, f) is a retraction pair. If $(f : A \to B, g : B \to A)$ is an m–embedding pair then

$$f(\perp_A) = \perp_B \text{ and } g(\perp_B) = \perp_A .$$

It also is easy to verify that if $(f : A \to B, g : B \to A)$ is a monotonic isomorphism pair, then both f and g are continuous.

The collection of cpo's and that of $m(c)$–projection pairs form a category that is denoted by $\mathcal{M}(\mathcal{C})$-$\mathcal{PCPO}$. $\mathcal{M}(\mathcal{C})$-$\mathcal{PCPO}$ is a non–full subcategory of $\mathcal{M}(\mathcal{C})\mathcal{CPO}$.

Function Spaces of Cpo's. Suppose A and B are cpo's. For every monotonic functions $f, g : A \to B$, we define a relation $f \leq g$ by

$$f \leq g \text{ iff } f(a) \leq g(a).$$

This relation clearly is a partial order over the set $[A \to B]_{\mathcal{M}}$ of all monotone functions from A to B. The restriction of this relation to the continuous functions is a partial order over the set $[A \to B]_{\mathcal{C}}$ of all continuous functions from A to B. For a poset A and a cpo B, $[A \to B]_{\mathcal{M}}$ and $[A \to B]_{\mathcal{C}}$ are cpo's.

Assume $(f_i : A_i \to B_i, f_i^R : B_i \to A_i),\ i = 1, 2$ are $m(c)$–projection pairs. They induce an $m(c)$–projection pair,

$$([f_1 \to f_2] : [A_1 \to A_2] \to [B_1 \to B_2], \quad [f_1^R \to f_2^R] : [B_1 \to B_2] \to [A_1 \to A_2]),$$

given by,

$$[f_1 \to f_2](w) = f_2 \cdot w \cdot f_1^R$$
$$[f_1^R \to f_2^R](w) = f_2^R \cdot w \cdot f_1.$$

Suppose A and B are finite cpo's; then all monotonic functions from A to B are continuous and

$$[A \to B]_{\mathcal{M}} = [A \to B]_{\mathcal{C}}.$$

The Inverse Limit Construction. By a *sequence of c–embeddings of cpo's*, we mean the following diagram:

$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{n-1}} A_n \xrightarrow{f_n} A_{n+1} \xrightarrow{f_{n+1}} \cdots \qquad \text{(Diagram D)}$$

where $f_n, n \in \omega$ are c–embeddings from A_n to A_{n+1}. The *inverse limit* of this sequence is a cpo,

$$\{x \in \mathit{\Pi}_{i\in\omega} A_i \mid x_i = f_i^R(x_{i+1})\},$$

with the point–wise ordering. We denote this cpo by A_∞. For each $n \in \omega$, we clearly have $\bot_{A_n} = f_n^R(\bot_{A_{n+1}})$ and $\bot_{A_\infty}$ is given as

$$\bot_{A_\infty} = \langle \bot_{A_0}, \bot_{A_1}, ...\rangle.$$

For each $n \in \omega$, there is a c–embedding pair $(f_{n\infty}, f_{n\infty}^R)$ from A_n to A_∞ such that

$$f_{n\infty}(x)(i) = \begin{cases} f_i^R \cdot f_{i+1}^R \cdot \ldots \cdot f_n^R(x) & \text{if } i < n \\ x & \text{if } i = n \\ f_n \cdot \ldots \cdot f_i(x) & \text{if } i > n, \end{cases}$$

$$f_{n\infty}^R(x) = x_n.$$

For each directed subset X of A_∞, the least upper bound of X is

$$\bigvee X = \bigvee_{n\in\omega} f_{n\infty}(\bigvee f_{n\infty}^R(X)).$$

The following naturally expected lemma states that $f_{n\infty}^R(x)$ is the n–th approximation of $x \in A_\infty$.

Proposition 1. *$\{f_{n\infty} \cdot f_{n\infty}^R(x) \mid n \in \omega\}$ is a chain in A_∞. Moreover,*

$$x = \bigvee\{f_{n\infty} \cdot f_{n\infty}^R(x) \mid n \in \omega\}.$$

We may write $\tilde{x}_n$ to denote $f_{n\infty} \cdot f_{n\infty}^R(x) = f_{n\infty}(x_n)$. Hence,

$$x = \bigvee\{\tilde{x}_n \mid n \in \omega\} = \bigvee_{n\in\omega} \tilde{x}_n.$$

Each element b of $\cup\{f_{n\infty}(A_n) \mid n \in \omega\}$ is *"compact"* since for every directed subset S of A_∞, if $b \leq \bigvee S$ then $b \leq s$ for some $s \in S$. Moreover, for every element s of A_∞, there is a directed set S_s of compact elements of A_∞ such that $s = \bigvee S_s$. Hence A_∞ is an *"algebraic cpo"*.

Maximal Elements of Cpo's Every chain is a directed set, whence it follows from (an equivalent to) the Axiom of Choice that every cpo has at least one maximal element. Indeed:

Lemma 2. *Let D be a cpo. Any element of D extends into a maximal element of D.*

Proof. Let $d \in D$ and let $U_d = \{x \in D \mid d \leq x\}$. Every chain in U_d is clearly a chain in D. Hence it has a least upper bound l in D. Clearly $l \in U_d$. Suppose there is an upper bound u of this chain in U_d such that

$$u \leq l.$$

Then, since $u \in D$, we have $u = d$. Hence, l is a least upper bound of this chain in U_d. By (an equivalent to) The Axiom of Choice, U_d has a maximal element m with respect to the induced partial ordering. Assume there is an element x of D such that

$$x \geq m.$$

Then,

$$x \geq m \geq d$$

and $x \in U_d$. Hence, $x = m$. Therefore this m is maximal with respect to the partial ordering on D.

Lemma 3. *Let A and B be cpo's such that $A \lhd_{\mathcal{M}} B$. Then, for each maximal element a of A, there is a maximal element b of B such that*

$$a = r(b),$$

where r is the m–retraction map from B to A.

Proof. Since $r : B \to A$ is surjective, there is $x \in B$ such that $a = r(x)$. Let $m \in B$ be a maximal extension of x. Then, due to the maximality of a and monotonicity of r, we have

$$r(m) = r(x) = a.$$

Lemma 4. *(The Isolation Lemma) Suppose A and B are nontrivial cpo's and consider the cpo $[A \to B]_{\mathcal{M}}$. For every maximal element f of $[A \to B]_{\mathcal{M}}$ and for all maximal elements a of A, $f(a)$ is maximal in B. In particular,*

$$f(a) \neq \perp .$$

Proof. Suppose for some $a \in A_{max}$, $f(a)$ is not maximal in B. Define $f' : A \to B$ by

$$f'(x) = \begin{cases} b & \text{if} \quad x = a \\ f(x) & \text{else,} \end{cases} \qquad (x \in A),$$

where b is an element of B with $b \gneq f(a)$. Since a is maximal, f' is monotone and hence $f' \in [A \to B]_{\mathcal{M}}$. Clearly $f' \gneq f$ and this is a contradiction.

This result states that maximal elements of cpo's are isolated with respect to monotonic functional application.

4.2 The Theory of SFP Objects

SFP objects were first studied by Plotkin [47] and Smyth [59] in order to build mathematical models of nondeterministic computation. Notwithstanding the original motivation, one finds that these objects are very important from the foundational point of view. The construction of SFP objects suggests that they are the smallest transfinite structures which bridge the finite world with the infinite world. The intellect's leap from finite to infinite goes through this transfinite transition between finitude and full scale actual infinity.

Consider a sequence of c–embeddings of cpo's,

$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{n-1}} A_n \xrightarrow{f_n} A_{n+1} \xrightarrow{f_{n+1}} \cdots \qquad \text{(Diagram D)}$$

where $f_n, n \in \omega$ are c–embeddings from A_n to A_{n+1}. Let A_∞ be the inverse limit of this sequence, i.e.,

$$A_\infty = \{x \in \Pi_{i \in \omega} A_i \mid x_i = f_i^R(x_{i+1})\}$$

with the point–wise ordering. In case, in the diagram D, all A_i, $i \in \omega$ are finite cpo's, we call the inverse limit A_∞ an *SFP object.* For any SFP object A_∞, $\cup\{f_{n\infty}(A_n) \mid n \in \omega\}$ is a countable set. Hence, A_∞ is an *"ω-algebraic cpo"*. We denote the category of SFP objects and monotonic (continuous) functions by $\mathcal{M}(\mathcal{C})\mathcal{SFP}$.

We consider the following sequences of c–embeddings of finite cpo's,

$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{n-1}} A_n \xrightarrow{f_n,} A_{n+1} \xrightarrow{f_{n+1}} \cdots$$

$$B_0 \xrightarrow{g_0} B_1 \xrightarrow{g_1} \cdots \xrightarrow{g_{n-1}} B_n \xrightarrow{g_n,} B_{n+1} \xrightarrow{g_{n+1}} \cdots$$

Now consider the following sequence of c–embeddings of finite cpo's,

$$C_0 \xrightarrow{h_0} C_1 \xrightarrow{h_1} \cdots \xrightarrow{h_{n-1}} C_n \xrightarrow{h_n,} C_{n+1} \xrightarrow{h_{n+1}} \cdots$$

where $C_n = [A_n \to B_n]_\mathcal{M}$ and

$$\begin{aligned} h_n(w) &= [f_n \to g_n](w) = g_n \cdot w \cdot f_n^R \\ h_n^R(w) &= [f_n^R \to g_n^R](w) = g_n^R \cdot w \cdot f_n. \end{aligned}$$

It is clear that $C_n, n \in \omega$ are finite cpo's (in fact, they are sets of monotone functions from A_n to B_n), and hence C_∞ also is an SFP object.

Proposition 2. *Let A_∞, B_∞, and C_∞ be as above. We have*

$$C_\infty \approx_{\mathcal{CCPO}} [A_\infty \to B_\infty]_\mathcal{C}.$$

Since treating this isomorphism explicitly is notationally cumbersome, we identify $[A_\infty \to B_\infty]_\mathcal{C}$ and C_∞. *To within this isomorphism,* each continuous function $u \in [A_\infty \to B_\infty]_\mathcal{C}$ can be considered as the limit of the chain $\{\tilde{u}_n\}_{n\in\omega}$, where

$$\tilde{u}_n = h_{n\infty} \cdot h^R_{n\infty}(u).$$

More formally,

$$u = \bigvee_{i\in\omega} \tilde{u}_i.$$

Furthermore, for each $n \geqslant 1$, we have $u_n = h^R_{n\infty}(u) = g^R_{n\infty} \cdot u \cdot f_{n\infty}$ and $\tilde{u}_n = h_{n\infty}(u_n)$.

Now suppose u is a non–continuous monotonic function from A_∞ to B_∞. Then, $g^R_{n\infty} \cdot u \cdot f_{n\infty}$ is a monotonic function from A_n to B_n; hence it still is an element of C_n. Note that $u \notin [A_\infty \to B_\infty]_\mathcal{C}$ whence u_n is not defined. However, for each $n \in \omega$

$$\begin{aligned}
&(h_{(n+1)\infty}(g^R_{(n+1)\infty} \cdot u \cdot f_{(n+1)\infty}))(x)\\
&\quad = g_{(n+1)\infty} \cdot g^R_{(n+1)\infty} \cdot u \cdot f_{(n+1)\infty} \cdot f^R_{(n+1)\infty}(x)\\
&\quad \geq g_{(n+1)\infty} \cdot g^R_{(n+1)\infty} \cdot u \cdot f_{n\infty} \cdot f^R_{n\infty}(x)\\
&\quad \geq g_{n\infty} \cdot g^R_{n\infty} \cdot u \cdot f_{n\infty} \cdot f^R_{n\infty}(x)\\
&\quad = (h_{n\infty}(g^R_{n\infty} \cdot u \cdot f_{n\infty}))(x)
\end{aligned}$$

where $x \in A_\infty$. Hence,

$$\{h_{n\infty}(g^R_{n\infty} \cdot u \cdot f_{n\infty}) \mid n \in \omega\}$$

still is a chain in C_∞.

Theorem 2. *(Maximum Continuous Approximation Theorem) Let A_∞ and B_∞ be SFP objects. Suppose u is a monotonic function from A_∞ to B_∞. Then,*

$$\bigvee\{h_{n\infty}(g^R_{n\infty} \cdot u \cdot f_{n\infty}) \mid n \in \omega\} \leq u.$$

Furthermore, u is continuous iff equality holds. Moreover, u has a largest continuous function,

$$\bigvee\{h_{n\infty}(g^R_{n\infty} \cdot u \cdot f_{n\infty}) \mid n \in \omega\},$$

that approximates u.

This theorem indicates that, over SFP objects, monotonic functions and continuous functions are very closely related. This theme is developed in the forthcoming section on hyper continuous functions of SFP objects.

Maximal Elements of SFP Objects.

Theorem 3. *Consider an SFP object A_∞ that is the inverse limit of the following sequence of c–embeddings of finite cpo's,*

$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \cdots \xrightarrow{f_{n-1}} A_n \xrightarrow{f_n,} A_{n+1} \xrightarrow{f_{n+1}} \cdots .$$

Then, every compact element of A_∞ extends into a maximal element.

Proof. Let e be a compact element of A_∞. Then, for some $k \in \omega$ and $e_k \in A_k$,

$$e = f_{k\infty}(e_k).$$

Let m_k be a maximal extension of e_k in A_k. For each $j \geqslant k$, let m_{j+1} be a maximal element of A_{j+1} such that

$$m_j = f_j(m_{j+1}).$$

Now, consider the following infinite tuple m,

$$m = < f_0^R \cdot ... \cdot f_{k-1}^R(m_k), ..., f_{k-1}^R(m_k), m_k, m_{k+1}, m_{k+2}, ... > .$$

It is clear that $m \in A_\infty$. Furthermore, $e \leq m$. Now assume $x \geq m$ in A_∞. Then, for all $j \geqslant k$,

$$f_{j\infty}^R(x) \geq f_{j\infty}^R(m) = m_j.$$

By the maximality of m_j, we have,

$$f_{j\infty}^R(x) = m_j.$$

Also for all $j < k$, we have,

$$f_{j\infty}^R(x) \geq f_{j\infty}^R(m) = f_j^R \cdot ... \cdot f_{k-1}^R(m_k),$$

and by the maximality of $f_0^R \cdot ... \cdot f_{k-1}^R(m_k)$, we have,

$$f_{j\infty}^R(x) = f_j^R \cdot ... \cdot f_{k-1}^R(m_k).$$

Therefore $x = m$. Hence, m is maximal in A_∞.

Remark 1. Note that this extension result does not use the Axiom of Choice. This indicates the intrinsic difference between the concept of actual infinity, e.g., infinite cpo's, and the transcendental infinity of SFP objects. The former requires Choice to reach maximal elements while the latter does not. This seems to be a counterexample to Cantor's fundamental postulate that the nature of infinite objects is a uniform generalization of that of finite ones.

SFP Solutions to Recursive Domain Equations. As a standard result, there is a non–trivial SFP object D_∞ such that

$$D_\infty \approx_{\mathcal{CSFP}} [D_\infty \to E]_{\mathcal{C}},$$

where E is an SFP object. We present an outline of the construction of such D_∞. Define finite cpo's recursively as follows,

$$D_0 = \bigcirc,$$

where $\bigcirc$ is a trivial cpo with sole element $\perp_\bigcirc$, and,

$$D_{n+1} = [D_n \to E]_{\mathcal{M}}.$$

Define $u_0 : D_0 \to D_1$ and $u_0^R : D_1 \to D_0$ as

$$u_0(\bot_{\bigcirc}) = \bot_{D_1} \text{ and } u_0^R(x) = \bot_{\bigcirc} .$$

Define $u_{n+1} : D_n \to D_{n+1}$ and $u_{n+1}^R : D_{n+1} \to D_n$ as

$$\begin{aligned} u_{n+1}(w) &= [u_n \to 1_E](w) = 1_E \cdot w \cdot u_n^R = w \cdot u_n^R \\ u_{n+1}^R(w) &= [u_n^R \to 1_E^R](w) = 1_E^R \cdot w \cdot u_n = w \cdot u_n. \end{aligned}$$

It is clear that (u_0, u_0^R) is a c-projection pair. Also $(1_E, 1_E^R) = (1_E, 1_E)$ is a c-projection pair. Therefore, (u_n, u_n^R) is a c-projection pair for all $n \in \omega$. Since D_n are all finite cpo's, D_∞ is an SFP object.

Clearly, for each $x, y \in D_\infty$ and for each $n \in \omega$, we have

$$x_{n+1}(y_n) \in E_n = E.$$

Also, it can readily be seen that $x_{n+1}(y_n)$, $n \in \omega$ forms a chain in E. Let

$$\Gamma : D_\infty \to [D_\infty \to E]_{\mathcal{C}}$$

be such that

$$\Gamma(x)(y) = \bigvee_{n \in \omega} x_{n+1}(y_n).$$

Then, Γ is a continuous function. Now, define

$$\Omega : [D_\infty \to E]_{\mathcal{C}} \to D_\infty$$

by

$$\Omega(v) = \bigvee_{n \in \omega} f_{n\infty}(v_{(n)}).$$

Then, Ω is a continuous function. Moreover,

$$\Gamma \cdot \Omega = 1_{[D_\infty \to E]_{\mathcal{C}}} \text{ and } \Omega \cdot \Gamma = 1_{D_\infty}.$$

4.3 Hyper-continuous Functions

The theory of maximal continuous approximations gives rise to a curious subclass of monotonic functions. Suppose A and B are SFP objects. A subspace $\lessdot A \to B \gtrdot \subseteq [A \to B]_{\mathcal{M}}$ is said to be *normal* iff for all $m \in [A \to B]_{\mathcal{M}}, c_m \in \lessdot A \to B \gtrdot$ where c_m is the maximal continuous approximation of m.

Trivially, $[A \to B]_{\mathcal{M}}$ is normal. Suppose m is an element of $[A \to B]_{\mathcal{C}}$. Then $c_m = m \in [A \to B]_{\mathcal{C}}$. Thus $[A \to B]_{\mathcal{C}}$ also is normal. Clearly, $[A \to B]_{\mathcal{C}}$ is the smallest normal space and $[A \to B]_{\mathcal{M}}$ is the largest normal space. Moreover, any subspace $\mathcal{F}$ such that

$$[A \to B]_{\mathcal{C}} \subseteq \mathcal{F} \subseteq [A \to B]_{\mathcal{M}}$$

is a normal space. *In conclusion, the normal spaces are precisely subspaces in between* $[A \to B]_{\mathcal{C}}$ *and* $[A \to B]_{\mathcal{M}}$, *inclusive.*

Suppose $\lessdot A \to B \gtrdot$ is normal and X is a cpo. A monotonic function $f : \lessdot A \to B \gtrdot \to X$ is said to be *hyper continuous* if for every $m \in \lessdot A \to B \gtrdot, f(m) = f(c_m)$. In words, *hyper continuous functions are those monotone functions which can not distinguish* m *from* c_m. We denote the set of all hyper continuous functions from $\lessdot A \to B \gtrdot$ to X by $\prec \lessdot A \to B \gtrdot \to X \succ$.

In the following, we study some technical but elementary properties of hyper continuous functions.

Lemma 5. *Let* $\lessdot A \to B \gtrdot$ *be normal and* X *be a cpo, then* $\prec \lessdot A \to B \gtrdot \to X \succ$ *is a cpo.*

Proof. Let $\divideontimes : \lessdot A \to B \gtrdot \to X$ be such that $\divideontimes(f) = \perp$ for all $f \in \lessdot A \to B \gtrdot$. Suppose $\divideontimes(m) = x$. Clearly $x = \perp$, and hence $\divideontimes(c_m) = x$. Hence $\divideontimes$ is the smallest hyper continuous function from $\lessdot A \to B \gtrdot$ to X. Suppose D is a directed subset of $\prec \lessdot A \to B \gtrdot \to X \succ$. Clearly it is a directed subset of $[\lessdot A \to B \gtrdot \to X]_{\mathcal{M}}$. Let δ be the least upper bound of D in $[\lessdot A \to B \gtrdot \to X]_{\mathcal{M}}$. It suffice to show that δ is hyper continuous. Suppose $m \in \lessdot A \to B \gtrdot$. For each $d \in D$, we have $d(m) = d(c_m)$. Hence,

$$\delta(m) = \bigvee_{d \in D} d(m) = \bigvee_{d \in D} d(c_m) = \delta(c_m).$$

We have established that δ is hyper continuous.

Lemma 6. *Suppose* $f : \lessdot A \to B \gtrdot \to X$ *is hyper continuous. Moreover, assume* f *is 1-1. Then* $\lessdot A \to B \gtrdot = [A \to B]_{\mathcal{C}}$.

Proof. Let $m \in \lessdot A \to B \gtrdot$. We have $f(m) = f(c_m)$. Since f is 1-1, $m = c_m$. Hence $\lessdot A \to B \gtrdot = [A \to B]_{\mathcal{C}}$.

Corollary 2. *Let* $\lessdot A \to B \gtrdot \neq [A \to B]_{\mathcal{C}}$. *Moreover, assume* $f : \lessdot A \to B \gtrdot \to X$ *is hyper continuous. Then,* f *is not injective.*

This Corollary establishes the following important result:

Theorem 4. *There are continuous functions which are not hyper continuous.*

Proof. Suppose $\lessdot A \to B \gtrdot$ is normal and $\lessdot A \to B \gtrdot \neq [A \to B]_{\mathcal{C}}$. Moreover $\lessdot A \to B \gtrdot$ is a cpo. Consider the identity function $id_{\lessdot A \to B \gtrdot}$ over $\lessdot A \to B \gtrdot$. It is injective and clearly continuous. Also $\lessdot A \to B \gtrdot$ is a cpo. Hence f is not hyper continuous.

Despite this general result, under a special circumstance, continuous functions become hyper continuous. Indeed,

Lemma 7. *We have*

$$\prec [A \to B]_{\mathcal{C}} \to X \succ = [[A \to B]_{\mathcal{C}} \to X]_{\mathcal{M}}.$$

Proof. For all $m \in [A \to B]_{\mathcal{C}}$, we have $m = c_m$. Hence all monotonic functions from $[A \to B]_{\mathcal{C}}$ to X are hyper continuous.

This also implies that whenever the normal space $\lessdot A \to B \gtrdot$ is an SFP object then all continuous functions from $\lessdot A \to B \gtrdot$ to a cpo X are hyper continuous. Furthermore, this lemma implies the following important result,

Corollary 3. *There are hyper continuous functions which are not continuous.*

Proof. Let $f : [A \to B]_{\mathcal{C}} \to X$ be a monotone function which is not continuous. This is hyper continuous but not continuous.

Corollary 4. *Suppose A, B and X are SFP objects; Then,*

$$\prec [A \to B]_{\mathcal{C}} \to X \succ \curvearrowleft_{\mathcal{M}} [[A \to B]_{\mathcal{C}} \to X]_{\mathcal{C}}.$$

Proof. By the maximal continuous approximation theorem, we have

$$[[A \to B]_{\mathcal{C}} \to X]_{\mathcal{C}} \curvearrowright_{\mathcal{M}} [[A \to B]_{\mathcal{C}} \to X]_{\mathcal{M}}.$$

In what follows, we present a general method of forcing monotone functions into hyper continuous functions and their relations.

Theorem 5. *Assume $\lessdot A \to B \gtrdot$ is normal, X is a cpo and f is a monotone function from $\lessdot A \to B \gtrdot$ to X. Then, there is a hyper continuous function $\bar{f} : \lessdot A \to B \gtrdot \to X$ such that $\bar{f} \leq f$. If f is hyper continuous then $f = \bar{f}$.*

Proof. Define $\bar{f} : \lessdot A \to B \gtrdot \to X$ by

$$\bar{f}(x) = \begin{cases} f(x) & \text{if } x \text{ is continuous} \\ f(c_x) & \text{if } x \text{ is monotone.} \end{cases}$$

This is to say $\bar{f}(x) = f(c_x)$. Suppose $x, y \in \lessdot A \to B \gtrdot$ and $x \leq y$. Let $\tilde{x}_i = h_{i\infty}(g^R_{i\infty} \cdot x \cdot f_{i\infty})$ and $\tilde{y}_i = h_{i\infty}(g^R_{i\infty} \cdot y \cdot f_{i\infty})$. Then,

$$c_x = \bigvee_{i \in \omega} \tilde{x}_i, \quad c_y = \bigvee_{i \in \omega} \tilde{y}_i$$

and, $\tilde{x}_i \leq \tilde{y}_i$ for all $i \in \omega$. Hence, $c_x \leq c_y$ and so

$$\bar{f}(x) = f(c_x) \leq f(c_y) = \bar{f}(y).$$

We have shown that $\bar{f}$ is monotone. Moreover,

$$\bar{f}(x) = f(c_x) = f(c_{c_x}) = \bar{f}(c_x).$$

Hence $\bar{f}$ is hyper continuous. Clearly $\bar{f} \leq f$. If f is hyper continuous, then

$$\bar{f}(x) = f(c_x) = f(x).$$

Corollary 5. *Assume $\lessdot A \to B \gtrdot$ is normal and X is a cpo. We have*

$$[\lessdot A \to B \gtrdot \ \to X]_{\mathcal{M}} \curvearrowleft_{\mathcal{MCPO}} \prec \lessdot A \to B \gtrdot \ \to X \succ .$$

Proof. Define

$$i : [\lessdot A \to B \gtrdot \ \to X]_{\mathcal{M}} \to \prec \lessdot A \to B \gtrdot \ \to X \succ$$

by

$$i(f) = \bar{f}.$$

Let $j : \prec \lessdot A \to B \gtrdot \ \to X \succ \to [\lessdot A \to B \gtrdot \ \to X]_{\mathcal{M}}$ be inclusion. Clearly both of them are monotone. We have $j \cdot i(f) = \bar{f} \leq f.$ and $i \cdot j(f) = i(f) = f.$

On the other hand, from the maximal continuous approximation theorem it follows that when X is an SFP object and $\lessdot A \to B \gtrdot \ = [A \to B]_{\mathcal{C}}$, for every hyper continuous function $f : \ \lessdot A \to B \gtrdot \ \to X$, there is a continuous function $c_f : \bar{f} : \ \lessdot A \to B \gtrdot \ \to X$ such that $c_f \leq f$, and such forcing constitutes the following embedding,

$$\begin{aligned}[\lessdot A \to B \gtrdot \ \to X]_{\mathcal{C}} \curvearrowright_{\mathcal{MCPO}} \prec \lessdot A \to B \gtrdot \ \to X \succ \\ = [\lessdot A \to B \gtrdot \ \to X]_{\mathcal{M}}.\end{aligned}$$

For normal spaces $\lessdot A_i \to B_i \gtrdot$, $i = 1, 2, ...n$, and a cpo X; we can define the concept of *hyper continuous n–ary functions*,

$$f : \ \lessdot A_1 \to B_1 \gtrdot \ \times \cdots \times \ \lessdot A_n \to B_n \gtrdot \ \to \ T$$

as monotone functions such that

$$f(x_1, ..., x_n) = f(c_{x_1}, ..., c_{x_n}).$$

Similar embedding results as for binary and multi–ary cases in general hold.

Suppose $\lessdot A \to B \gtrdot$ and $\prec \lessdot A \to B \gtrdot \ \to X \succ$ are normal. Define a function $Ap : \prec \lessdot A \to B \gtrdot \ \to X \succ \times \lessdot A \to B \gtrdot \ \to X$ as

$$Ap(f, d) = c_f(c_d).$$

Lemma 8. *Ap is hyper continuous.*

Proof. We have

$$Ap(f, d) = c_f(c_d) = c_{c_f}(c_{c_d}) = Ap(c_f, c_d).$$

Hence Ap is hyper continuous. This can be proved alternatively as follows: The function ap given by

$$ap(f, d) = f(d)$$

is continuous. Hence $Ap = \overline{ap}$ is hyper continuous.

4.4 Balancing a Hyper-continuous Function Space with Its Domain

We have seen that by restricting the members of a function space to continuous functions, we can balance the size of the space of functions with the size of its domain in the category $\mathcal{CSFP}$ of continuous functions over SFP objects. In what follows, we show that this balancing of size can be achieved in the category $\mathcal{MCPO}$ of monotone functions over cpo's.

Suppose D_∞ is a solution to

$$D \approx_{\mathcal{CSFP}} [D \to E]_{\mathcal{C}}$$

in $\mathcal{CSFP}$. Hence we can identify D_∞ with $[D_\infty \to E]_{\mathcal{C}}$. Precisely speaking, we ought to be explicit about the isomorphism pair for $D_\infty \approx_{\mathcal{CSFP}} [D_\infty \to E]_{\mathcal{C}}$. However, to avoid notational complexity, we identify $d \in D_\infty$ with the isomorphic image of d, unless doing so causes confusion. Let $M = \prec D_\infty \to E \succ$; then $M = [D_\infty \to E]_{\mathcal{M}}$. Hence,

$$D_\infty \approx_{\mathcal{CCPO}} [D_\infty \to E]_{\mathcal{C}} \curvearrowright_{\mathcal{M}} [D_\infty \to E]_{\mathcal{M}} = M,$$

and we have

$$[D_\infty \to E]_{\mathcal{C}} \curvearrowright_{\mathcal{M}} M.$$

Furthermore,

$$D_\infty \curvearrowright_{\mathcal{M}} M.$$

To within the identification of D_∞ with $[D_\infty \to E]_{\mathcal{C}}$, this embedding is the inclusion map ic and the adjoint projection is the monotone map

$$c : m \mapsto c_m.$$

Notice that, in general, M is a cpo but is not an SFP object.

Theorem 6. $M \approx_{\mathcal{MCPO}} \prec M \to E \succ$.

Proof. Let

$$(\beta : [D_\infty \to E]_{\mathcal{C}} \to M, \alpha : M \to [D_\infty \to E]_{\mathcal{C}})$$

be the embedding pair for

$$[D_\infty \to E]_{\mathcal{C}} \curvearrowright_{\mathcal{M}} M.$$

So, $\alpha(m) = c_m$ and β is the inclusion $ic : u \mapsto u$. Define

$$\Phi : M \to \prec M \to E \succ \ \text{ and } \Psi : \prec M \to E \succ \to M$$

by

$$\Phi(a) = a \cdot \alpha \quad \Psi(b) = b \restriction [D_\infty \to E]_{\mathcal{C}}.$$

If for $m \in M$ and compact $x \in E$, $(\Phi(a))(m) \geq x$, then

$$(\Phi(a))(c_m) = a(c_{c_m}) = a(c_m) = (\Phi(a))(m) \geq x.$$

Hence $\Phi(a)$ is hyper continuous and is indeed in $\prec M \rightarrow E \succ$. Moreover, for $a \geq a'$, and $m \in M$, we have,

$$\Phi(a)(m) = a(c_m) \geq a'(c_m) = \Phi(a')(m).$$

Hence Φ is monotone. For all $d \in D_\infty$, we have,

$$(\Psi(b))(d) = b(d),$$

and, since $b \in \prec M \rightarrow E \succ$ is monotone, $\Psi(b)$ is monotone. So, $\Psi(b)$ indeed is in M. Moreover, if $b \geq b'$ and then

$$(\Psi(b))(d) = b(d) \geq b'(d) = (\Psi(b'))(d).$$

Hence Ψ is monotone. We have seen that Φ and Ψ are both well–defined. Now we verify that they constitute an isomorphism pair. For each $b \in \prec M \rightarrow E \succ$ and $m \in M$, we have,

$$(\Phi \cdot \Psi(b))(m) = (\Psi(b) \cdot \alpha)(m) = b(c_m) = b(m).$$

So, $\Phi \cdot \Psi = 1_{\prec M \rightarrow E \succ}$. Moreover, for each $a \in M$ and $d \in D_\infty$, we have.

$$(\Psi \cdot \Phi(a))(d) = \Psi(a \cdot \alpha)(d) = a \cdot \alpha(d) = a(c_d) = a(d).$$

Therefore, $\Psi \cdot \Phi = 1_M$.

The following immediate consequence of this isomorphism holds.

Lemma 9. *For all $m \in M$, we have,*

$$\forall x \in D_\infty.[m(x) \leq m'(x)] \textit{ iff } \forall u \in M.[\Phi(m)(u) \leq \Phi(m')(u)].$$

We may write $m \prec u \succ$ for $\Phi(m)(u)$. To be specific, we will denote the partial ordering on M by $\leq$ and that on $\prec M \rightarrow E \succ$ by $\preceq$. Therefore this lemma states

$$m \leq m' \text{ iff } \Phi(m) \preceq \Phi(m').$$

Moreover, we have,

Theorem 7. *For all $d \in D_\infty$ and $m \in M$, $m \prec d \succ = m(c_d)$. For each $m \in M$, let $apply_m : M \rightarrow E$ be*

$$apply_m(x) = m \prec x \succ .$$

It is clear that $apply_m$ is hyper continuous. However, $apply : M \times M \rightarrow E$ defined as

$$apply((m, x)) = m \prec x \succ$$

is monotone but not hyper continuous.

4.5 The Construction of M_{max}

The construction of the canonical PFS M_{max} may now be given in three stages. First, using the theory of SFP objects, we construct a continuum reflexive structure D_∞ satisfying $D_\infty \approx_{\mathcal{CSFP}} [D_\infty \to T]_{\mathcal{C}}$, where T is the domain of three–valued truth,

$$\begin{tikzcd}
\underline{true} & & \underline{false} \\
& \bot \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

under the information ordering $\leq_k$ and $[D_\infty \to T]_{\mathcal{C}}$ is the space of all continuous functions from D_∞ to T under the information ordering associated with nested partial characteristic functions. Here, the bottom value $\bot$ represents a truth–value gap. The truth value domain $(T, \leq_k)$ is clearly an SFP object. In truth theory, this structure $(T, \leq_k)$ is used to construct term models of partial logic; the third truth value $\bot$ is successfully used as the semantic value of logically paradoxical sentences [28, 35, 20].

The reason for adopting $(T, \leq_k)$ as our domain of truth values is two– fold. First, it is an SFP structure and hence we can use it as a fixed parameter in recursive structural equations. Second, we can use $\bot$ to capture the behavior of logical paradoxes in set theory. There is another partial ordering associated with the underlying set T. This partial ordering, called the *truth ordering* and denoted by $\leq_t$, is described by the following diagram,

$$\begin{tikzcd}
\underline{true} \\
\bot \arrow[u] \\
\underline{false}. \arrow[u]
\end{tikzcd}$$

Following Kleene [33], the strong three–valued Boolean operators $\overline{\wedge}, \overline{\vee}$, and $\overline{\neg}$ are defined on T by,

$\overline{\wedge}$: the greatest lower bound operation with respect to $\leq_t$
$\overline{\vee}$: the least upper bound operation with respect to $\leq_t$
$\overline{\neg}$: $(\underline{true}) = \underline{false}$
$\overline{\neg}$: $(\underline{false}) = \underline{true}$
$\overline{\neg}$: $(\bot) = \bot$.

[7] introduces a complete set of truth functions for Kleene's strong 3–valued logic; here it suffices to observe that strong material conditional $\overline{\rightarrow}$ and biconditional $\overline{\leftrightarrow}$ are monotone operations over $(T, \leq_k)$ under their classical definitions in terms of strong negation and disjunction (or conjunction). The quantifiers $\overline{\forall}, \overline{\exists}$ are introduced in terms of arbitrary meet $\bigwedge$ and join $\bigvee$ on $(T, \leq_t)$. These connectives are monotone functions over $(T, \leq_k)$ but fail to be continuous due to their infinitary nature: D_∞ is not closed under $\bigwedge$ and $\bigvee$. This hampers D_∞ as

a first order model for naive set theory[4], as does the connected failure of strict identity to be even monotone as a binary function $D_\infty \times D_\infty \to T$.

However, we have seen that each monotone function $f : D_\infty \to T$ is maximally approximated by a unique continuous function c_f in $[D_\infty \to T]_{\mathcal{C}}$, whence c_f in D_∞ under representation. This gives rise to the concept of hyper continuity.

Next, we have seen that the space M of monotone functions from D_∞ to T is a solution for the reflexive equation $M \approx_{\mathcal{M}} \prec M \to T \succ$, where $\prec M \to T \succ$ is the space of all hyper continuous functions from M to T. Throughout the rest of this section, we consider a solution $M = [D_\infty \to T]_{\mathcal{M}}$ for $W \approx_{\mathcal{M}} \prec W \to T \succ$ as discussed in the previous section. Note that a monotone function $f : M \to T$ is hyper continuous just in case

$$c_x = c_y \;\Rightarrow\; f(x) = f(y) \qquad (x, y \in M),$$

i.e., over M, the equivalence relation of sharing a common maximal continuous approximation is a congruence for all hyper continuous functions. The isomorphism pair,

$$(\Phi : M \to \prec M \to T \succ,\ \Psi : \prec M \to T \succ \to M),$$

yields application $\cdot \prec \cdot \succ : M \times M \to T$ defined by

$$b(a) = \Phi(b)(a) \qquad (a, b \in M).$$

Any hyper continuous map $f : M \to T$ is represented by a unique object $\Psi(f)$ in M such that

$$\Psi(f)(a) = f(a).$$

Writing "$x \bar{\in} y$" for "$y \prec x \succ$ ", we pass to a universe $(M, \bar{\in})$ of partial (non–bivalent) sets–in–extension similar to the set theories of Gilmore [28] and Scott [57]. Let $x \equiv_M y =_{df} (\forall m \in M)(m \bar{\in} x \overline{\Leftrightarrow} m \bar{\in} y)$. $(M, \equiv_M)$ is thus an equivalence relation and a congruence for all hyper continuous functions from M to T. Although $x \equiv_M y$ is hyper continuous in both x and y, the hyper continuity of indiscernibility will be established only in the context of the identity theory of *maximal* elements of M, to which we now pass.

Maximal Elements of *M*. Since $(M, \leq)$ is a complete partial order, M is closed under least upper bounds of $\leq$–chains. Hence, there are $\leq$–maximal elements of M by (an equivalent to) the Axiom of Choice. Let M_{max} be the set of maximal elements of M.

Theorem 8. *Let $a, b \in M_{max}$. Then $a \prec b \succ \neq \perp$.*

[4] Historically, this failure of quantificational continuity surfaced in the functional setting of Church's quantified λ-calculus in the form of Rossers's Paradox [34], a functional version of Russell's Paradox.

Proof. Suppose $a \prec b \succ = a(c_b) = \perp$. Let $X = \{x \in D_\infty \mid c_b \leq x\}$. To show that X is a directed subset of D_∞, suppose that $d_1, d_2 \in X$ lack an upper bound in $(D_\infty, \leq)$. Then there is $d_3 \in D_{max}$ such that $d_1(d_3) = \underline{true}$ and $d_2(d_3) = \underline{false}$. Then, since $c_b \leq d_1, d_2$, we have $c_b(d_3) = \perp$. Since b is maximal in M and d_3 is maximal in D_∞, $b(d_3) \neq \perp$ by The Isolation Lemma. So we assume, without loss of generality, that $b(d_3) = \underline{true}$; the case $b(d_3) = \underline{false}$ is treated in a manner similar to the following: Since $d_1(d_3) = \underline{true}$ by the continuity of d_1 there is a compact approximation e to d_3 in D_∞, i.e., $e \leq d_3$, such that $d_1(e) = \underline{true}$. Define $f : D_\infty \to T$ by

$$f(d) = \begin{cases} \underline{true} \text{ if } & e \leq d \\ c_b(d) \text{ else,} & \end{cases} \qquad (d \in D_\infty).$$

To show that f is a function (i.e, it is single–valued), suppose there is a $d \in D_\infty$ such that $e \leq d$ and $f(d) = c_b(d) = \underline{false}$. Since $c_b \leq d_1$, by monotonicity $d_1(d) = \underline{false}$. But as $d_1(e) = \underline{true}$ and $e \leq d_1$, by monotonicity $d_1(d) = \underline{true}$, a contradiction. Thus, f is a function extending c_b, i.e., $c_b < f$. Indeed f is monotone by construction. Further, we claim f is continuous: let $\{y_\alpha\}$ be a directed subset of D_∞ and $y = \lim\{y_\alpha\}$; we need to show there is an ordinal α with $f(y) = f(y_\alpha)$. If $e \leq y$, then by the compactness of e, there is an ordinal $\beta \leq \alpha$ and $y_\beta \in \{y_\alpha\}$ such that $e \leq y_\beta$. Then,

$$f(y_\beta) = f(e) = f(y) = \underline{true}$$

with the first and second identity by the montonicity of f, and the last by the definition of f, whence y_β is the desired y_α. Otherwise, $f(y) = c_b(y)$ and the desired y_α is given by the continuity of c_b. This contradicts the maximality of c_b among continuous approximation to b. Thus, X is a directed subset of D_∞.

(Note that, as D_∞ is a cpo, X has a least upper bound $d^* \in D_\infty$, whence $d^* \in D_{max}$ since X is upward closed under $\leq$. By the Isolation Lemma, $a \prec d^* \succ \neq \perp$, since a is maximal in M and d^* is maximal in D_{max}.)

Define $g : D_\infty \to T$ by

$$g(d) = \begin{cases} \underline{true} \text{ if } & c_b \leq d \\ a(d) \text{ else,} & \end{cases} \qquad (d \in D_\infty).$$

Then g is a function as X is directed. f is monotone by construction. Further, $a < g$, contrary to a's maximality in $(M, \leq)$. Thus $a \prec b \succ \neq \perp$, as required.

We say $a \in M$ is *max–defined* iff for all maximal elements m of M, $a \prec m \succ \neq \perp$. By the previous Lemma, all elements of M_{max} are max–defined.

Lemma 10. *If $f : M \to T$ is hyper continuous, then $\Psi(f) \in M$ is max–defined iff for all $x \in M_{max}$, $f(x) \neq \perp$.*

Proof. Suppose $\Psi(f) \in M$ is max–defined. Then, for all $x \in M_{max}$,

$$f(x) = \Phi \cdot \Psi(f)(x) = \Psi(f) \prec x \succ \neq \perp .$$

Conversely, assume for all $x \in M_{max}$, $f(x) \neq \perp$. We then have

$$\Psi(f) \prec x \succ = \Phi \cdot \Psi(f)(x) = f(x) \neq \perp .$$

Hence $\Psi(f)$ is max–defined.

Theorem 9. *(The Semi–lattice Lemma) Suppose $a \in M$ is max–defined. Then the collection,*

$$\{b \in M \mid a \leq b\},$$

is an upper semi–lattice.

Corollary 6. *Every element of M which is max–defined has a unique maximal extension $m_a \in M$.*

Proof. Since M is a cpo, any $a \in M$ which is max–defined has a maximal extension. Let a' and a'' be maximal extensions of a. Then, by The Semi–Lattice Lemma, $a' \vee a''$ exists in M; whence $a' = a''$.

If a is max–defined, then the unique maximal extension m_a of a agrees with a on maximal elements. So, *intuitively, we can consider m_a the completion of a with respect to non–maximal arguments.*

There is a trivial but important consequence to this Corollary.

Theorem 10. *Suppose $m \in M$ is max–defined and $\mathbf{m}$ is the unique maximal extension of m. Then, for all $x \in M_{max}$*

$$\mathbf{m} \prec x \succ = m \prec x \succ = \Phi(m)(x).$$

Theorem 11. *Suppose $a', a'' \in M$ are max–defined and they agree on maximal elements of M. Then, they have the same maximal extension.*

Corollary 7. *Suppose $a, a' \in M$ are maximal and they agree on maximal elements of M. Then, $a = a'$.*

Max–indiscernibility If $a, a' \in M$ are such that for all $m \in M_{max}$,

$$m \prec a \succ = m \prec a' \succ,$$

then we say a and a' are *max–indiscernible*, in symbols $a \equiv_{max} b$. Obviously if $a = b$ then a and b are max–indiscernible. Moreover, $a, a' \in M$ are max–indiscernible iff for all maximal elements f of $\prec M \to T \succ$, $f(a) = f(b)$. It is easy to verify that $\equiv_{max}$ is an equivalence relation on M.

Theorem 12. *Suppose $a, a' \in M_{max}$ agree on maximal elements of M. Then, they are max–indiscernible.*

Proof. Immediate from Corollary 7.

The max–indiscernibility relation restricted to maximal elements of M can be represented by the following hyper continuous relation $\overline{\equiv}_{max} : M \times M \to T$,

$$a \,\overline{\equiv}_{max} b = \overline{\forall} m \in M_{max}.[m \prec a \succ \; \overline{\leftrightarrow}\, m \prec b \succ].$$

In fact, we have,

Lemma 11. *Let $a, b \in M_{max}$. Then, a and b are max–indiscernible iff $a \,\overline{\equiv}_{max}\, b = \underline{true}$. Moreover, a, b are* not *max–indiscernible iff $a \,\overline{\equiv}_{max}\, b = \underline{false}$.*

Moreover,

Lemma 12. *For all $a \in M$, $a \,\overline{\equiv}_{max}\, c_a = \underline{true}$.*

Proof. For all $m \in M_{max}$, due to the hyper continuity of $\Phi(m)$, we have

$$m \prec a \succ = \Phi(m)(a) = m(c_a) = \Phi(m)(c_a) = m \prec c_a \succ .$$

Due to the nature of $\overline{\forall}$ and $\overline{\leftrightarrow}$, we have $a \,\overline{\equiv}_{max}\, b \neq \perp$ iff for all $m \in M_{max}$, $m \prec a \succ \neq \perp$ and $m \prec b \succ \neq \perp$. The following lemma immediately follows,

Lemma 13. *Let $a, b \in M$. If $a \,\overline{\equiv}_{max}\, b = \perp$ then, one of a or b is not maximal.*

Proof. Suppose $a \overline{\equiv}_{max} b = \perp$. By the observation above, for some $m \in M_{max}$,

$$m \prec a \succ \neq \perp \quad \text{or} \quad m \prec b \succ \neq \perp .$$

Hence, either a or b is not maximal.

Remark 2. The converse of this result does not hold. Indeed, by Lemma 12, for all $a \in M_{max} - D_\infty, a \overline{\equiv}_{max} c_a \neq \perp$, but c_a is not maximal.

Theorem 13. *Suppose $m \in M$ is max–defined and $\mathbf{m}$ is the unique maximal extension of m. Then $\mathbf{m} \equiv_{max} m$.*

Max–Coextensiveness If $a \in M$ and $b \in M$ agree on all elements m of M_{max}, in the sense that

$$a \prec m \succ = b \prec m \succ,$$

then we say a and b are *max–coextensive*, in symbols $a \sim_{max} b$. This is equivalent to saying that, for all maximal elements x of D_∞,

$$a(x) = b(x).$$

It can readily be verified that $\sim_{max}$ is an equivalence relation on M. This relation restricted to max–defined elements of M has the following monotone (but not hyper continuous) representation,

$$a \,\overline{\sim}_{max}\, b = \overline{\forall} m \in M_{max}.[a \prec m \succ \; \overline{\leftrightarrow}\, b \prec m \succ].$$

Indeed, we have,

Lemma 14. *Suppose a and b are max–defined. Then, a and b are max–coextensive iff $a \mathrel{\overline{\eqsim}}_{max} b = \underline{true}$. Moreover, a and b fail to be max–coextensive iff $a \mathrel{\overline{\eqsim}}_{max} b = \underline{false}$.*

Due to the nature of $\overline{\forall}$ and $\overline{\Leftrightarrow}$, we have $a \mathrel{\overline{\eqsim}}_{max} b \neq \bot$ iff for all $m \in M_{max}, a \prec m \succ \neq \bot$ and $b \prec m \succ \neq \bot$. This establishes the following lemma,

Lemma 15. *Suppose $a, b \in M$. Either a or b is not max–defined iff*

$$a \mathrel{\overline{\eqsim}}_{max} b = \bot \ .$$

We call $\equiv_{max}$ and $\sim_{max}$ restricted to M_{max} the *indiscernibility* and the *co–extensiveness* relations over M_{max}, respectively.

Theorem 14. *Let $a, b \in M_{max}$. Then $a \sim_{max} b$ iff $a = b$.*

Theorem 15. *Let $a, b \in M_{max}$. Then $a \equiv_{max} b$ iff $c_a = c_b$.*

The indiscernibility relation $\equiv_{max}$ is thus coarser than co–extensiveness $\sim_{max}$, which is strict identity $=$ over M_{max} and the restriction of a monotone function from $M \times M$ to T to maximal elements of M.

$(M_{max}, \equiv_{max})$. $(M_{max}, \equiv_{max})$ is a "reduct" of $(M, \equiv_M)$ in the sense that it preserves the discriminative capacities of M : if a and $b \in M_{max}$ fail to be discerned by elements of M_{max}, then they fail to be discerned by elements of M,

$$a \ \equiv_{max} \ b \Leftrightarrow a \equiv_M b \qquad (a, b \in M),$$

a corollary of the theorem that, under $\leq$, every element a of M which is max–defined has a unique $\leq$ maximal extension m_a in M_{max} which agrees with a on all elements of M_{max} i.e., $a \sim_{max} m_a$.

A subset X of M_{max} is *exact* iff X is closed under $\equiv_{max}$. Throughout this section "X" ranges over exact subsets. We now establish that:

$$\mathcal{C}(\equiv_{max}) \approx M_{max} \lhd 2^{M_{max}}.$$

Let $\{M_{max} \to 2\}$ be the family of all characteristic functions f_X of exact subsets X of M_{max}. For each exact $X \subseteq M_{max}$, let $g(X) : M \to T$ be the hyper continuous function extending f_X by assigning the function value $\bot$ to all non maximal arguments in M,

$$g(X)(m) = \begin{cases} f(m) \text{ if } & m \in M_{max} \\ \bot & \text{else,} \end{cases} \qquad (m \in M).$$

Then $f_X \leq g(X)$. Since f is max–defined, $\Psi(g(X))$ has a unique maximal extension $\mathbf{\Psi}(\mathbf{g}(\mathbf{X})) \in M$.

Define $l : \{M_{max} \to 2\} \to M_{max}$ by

$$l(f_X) = \mathbf{\Psi}(\mathbf{g}(\mathbf{X})).$$

Adjoint–wise, define $r : M_{max} \to \{M_{max} \to 2\}$ by

$$r(m) = \Phi(m) \restriction M_{max} \qquad (m \in M_{max}).$$

Then $l(r(m)) = m$ $(m \in M_{max})$, whence (l, r) is an isomorphism pair of M_{max} and $\{M_{max} \to 2\}$. Let $\mathfrak{G} =_{df} (M_{max}, \equiv_{max}, \ulcorner\cdot\urcorner, \llcorner\cdot\lrcorner)$, where

$$\ulcorner.\urcorner : 2^{M_{max}} \to M_{max}, \qquad \llcorner\cdot\lrcorner : M_{max} \to 2^{M_{max}}$$

are defined by $\ulcorner Y \urcorner = l(Cl(Y))$ $(Y \in 2^{M_{max}})$, and $\llcorner m \lrcorner =$ the subset X of M_{max} whose characteristic function is $r(m)$ (i.e., such that $f_X = r(m)$) $(m \in M_{max})$.

Then $\mathfrak{G}$ is a PFS. The cardinality of M_{max} is that of $2^{\mathbb{R}}$ (hypercontinuum) but merely that of 2^{ω} (continuum) up to $\equiv_{max}$.

See [2] for a proof that $\mathfrak{G}$ is a plenum satisfying DoD.

4.6 Graded Indiscernibility in M_{max}

The Maximum Ccontinuous Approximation Theorem asserts that every $D_{n+1} = [D_n \to T]_{\mathcal{M}}$ induces the following indiscernibility E_{n+1} on M_{max}. Let $x, y \in M_{max}$ and let c_x, c_y be maximum continuous approximations of x, y respectively. Let d_x, d_y the projection of c_x, c_y to D_{n+1}. Then $E_{n+1} \subseteq M_{max} \times M_{max}$ is the equivalence relation defined by

$$xE_{n+1}y \Leftrightarrow d_x = d_y \ (x, y \in M_{max}).$$

Then $E_{n+1} \subseteq E_n$; $\{E_n\}$ is a descending chain of equivalence relations on M_{max} approximating $\equiv_{max}$. This is an interesting example of an “infinitary information system” in the sense of [51], to which we now turn.

5 Graded Indiscernibility and Metric

Polkowski [51] induced a graded indiscernibility relation over a universe U from an infinite sequence of attributes a_i via the conversion of an attribute a_i to an equivalence relation IND_a such that

$$IND_{a_i}(x, y) \Leftrightarrow a_i(x) = a_i(y).$$

Such sequences of attributes are called “infinite information systems”.

[51] considers a descending chain of equivalence relations $\{E_i\}$ over U such that $E_{i+1} \subseteq E_i$. We call the equivalence relation $E_\infty = \bigcap_{i \in \omega} E_i$ an *indiscernibility relation graded by* $\{E_i\}$. By $[x]_i$ we mean the equivalence class $\{y : xE_iy\}$. It is clear that

$$[x]_{n+1} \subseteq [x]_n.$$

A subset $T \subseteq U$ is *n–complete (or n–exact)* if

$$T = \bigcup\{[x]_n : x \in T\}.$$

Otherwise, T is said to be *n–rough.* For each n, the family,

$$\mathcal{B}_n = \{[x]_n : x \in U\},$$

is the partition of U induced by the equivalence relation E_n and it forms a base for a Polkowski-Pawlak topology Π_n over U. It is clear that open sets in this topology are precisely n–exact sets. Given a subset $T \subseteq U$, it is easy to verify that T is n–exact iff

$$Cl_n(T) = Int_n(T),$$

where $CL_n(T)$ and $INT_n(T)$ are closure and the interior in Π_n of T, respectively.

We have

$$[x]_n = \Big[\bigcup\{[z]_{n+1} : z \in [x]_n - [x]_{n+1}\}\Big] \cup [x]_{n+1}.$$

So, all (basic) open sets in Π_n are open in Π_{n+1}. That is Π_{n+1} *is finer than* Π_n. Similarly, all (basic) open sets in Π_n are open in Π_∞ and Π_∞ is finer than Π_n.

Having established that Π_∞ is an upper bound of the chain $\Pi_n \subseteq \Pi_{n+1}$, it is natural to "expect" that

$$\Pi_\infty = \bigcup_{n\in\omega} \Pi_n,$$

which we will examine in what follows.

Let

$$\mathcal{B}_{\mathcal{A}} = \bigcup_n \mathcal{B}_n = \{[x]_n : n \in \omega, x \in U\}.$$

It can readily be shown that $\mathcal{B}_{\mathcal{A}}$ forms a base for a topology $\Pi_{\mathcal{A}}$ over U. It is clear that all (basic) open sets of Π_n are open in $\Pi_{\mathcal{A}}$. So, $\Pi_{\mathcal{A}}$ is finer than Π_n.

It is clear that

$$\Pi_{\mathcal{A}} \supseteq \bigcup_{n\in\omega} \Pi_n$$

in lieu of the definition of $\mathcal{B}$. Conversely assume X is open in $\Pi_{\mathcal{A}}$. Then, for some $u \subseteq \omega$ and $w \subseteq U$,

$$X = \cup\mathcal{X},$$

where $\mathcal{X} = \{[x]_n : n \in u, x \in \omega\}$. For each n, let

$$\mathcal{X}_n = \{[x]_n : x \in w\}.$$

Then

$$\bigcup \mathcal{X}_n \in \Pi_n,$$

and

$$X = \bigcup_{n\in\omega} \bigcup \mathcal{X}_n \in \bigcup_{n\in\omega} \Pi_n.$$

We have shown that

$$\Pi_{\mathcal{A}} = \bigcup_{n\in\omega} \Pi_n.$$

So, $\Pi_{\mathcal{A}}$ is the least upper bound of the chain $\Pi_n \subseteq \Pi_{n+1}$, $n \in \omega$. It now follows that

$$\Pi_{\mathcal{A}} \subseteq \Pi_\infty.$$

This can be verified by the following direct argument too: As we have

$$[x]_n = \left[\bigcup\{[z]_\infty : z \in [x]_n - [x]_\infty\}\right] \cup [x]_\infty,$$

$[x]_n$ is open in Π_∞. Conversely,

$$[x]_\infty = \bigcap_{n\in\omega} [x]_n.$$

We know that $(\Pi_n, \subseteq)$ is a complete lattice, as $(\Pi_n, \cup, \cap, \neg, \varnothing, U)$ is a complete Boolean algebra, and hence,

$$\left(\bigcup_n \Pi_n, \subseteq\right)$$

is a complete lattice with the top element U and the bottom element $\varnothing$. Therefore, $\bigcap_{n\in\omega}[x]_n \in \bigcup_n \Pi_n$. We have shown that

$$\Pi_{\mathcal{A}} \supseteq \Pi_\infty.$$

In summary,

Proposition 3. $\Pi_{\mathcal{A}} = \bigcup_{n\in\omega} \Pi_n = \Pi_\infty$.

This allows us to introduce the following definition: $T \subseteq U$ is $\Pi_{\mathcal{A}}$*–exact* iff it is Π_∞-exact, i.e.,

$$T = \cup\{[x]_\infty : x \in T\}.$$

It immediately follows that T is $\Pi_{\mathcal{A}}$-exact iff

$$T = Cl_{\Pi_{\mathcal{A}}}(T) = Int_{\Pi_{\mathcal{A}}}(T),$$

where $Cl_{\Pi_{\mathcal{A}}}(T)$ and $Int_{\Pi_{\mathcal{A}}}(T)$ are the closure and the interior of T with respect to the topology $\Pi_{\mathcal{A}}$. A subset of U which is not $\Pi_{\mathcal{A}}$-exact is called $\Pi_{\mathcal{A}}$*–rough.*

For each $n \in \omega$, we define a function $d_n : U \times U \to \mathbf{R}^+$, where $\mathbf{R}^+$ is the collection of nonnegative real numbers, as follows,

$$d_n(x, y) = \begin{cases} 1 \text{ if } [x]_n \neq [y]_n \\ 0 \text{ otherwise.} \end{cases}$$

Then,

$$d_n(x, x) = 0, \quad d_n(x, y) = d_n(y, x).$$

We now define a function $d : U \times U \to \mathbf{R}^+$ as

$$d(x, y) = \sum_n 10^{-n} \cdot d_n(x, y),$$

where the convergence of the right side of the equation can readily be verified. It immediately follows that d is a pseudo–metric on U. [So is d_n.] This pseudo–metric d contains the information on when x and y became indiscernible in the chain E_i. In particular,

Lemma 16. *$d(x, y) = 0$ whenever $xE_\infty y$.*

Remember that as $(\Pi_\mathcal{A}, U, \varnothing, \cup, \cap, \neg)$ is a complete Boolean algebra, all open sets in $\Pi_\mathcal{A}$ are closed and vice versa. Now, according to Hausdorff's account, the following defines a metric on $\Pi_\mathcal{A}$:

$$d_H(K, L) = \max\{\max_{x\in K} dist(x, L), \max_{x\in L} dist(x, K)\},$$

where

$$dist(x, L) = \min\{d(x, z) : z \in L\}.$$

It is well–known that every $\Pi_\mathcal{A}$-rough set X can be characterized by a pair of $\Pi_\mathcal{A}$-exact sets (Q, T) where

$$Q = Cl_{\Pi_\mathcal{A}}(X), \quad T = U - Int_{\Pi_\mathcal{A}}(X).$$

Given two $\Pi_\mathcal{A}$-rough sets X_1, X_2, we can define the distance metric D by

$$D(X_1, X_2) = \max\{d_H(Q_1, Q_2), d_H(T_1, T_2)\}.$$

If X is $\Pi_\mathcal{A}$-exact, then,

$$Q = Cl_{\Pi_\mathcal{A}}(X) = X, \quad T = U - Int_{\Pi_\mathcal{A}}(X) = U - X.$$

It can readily be verified that

$$d_H(Q_1, Q_2) = d_H(X_1, X_2) = d_H(T_1, T_2).$$

Therefore, $D(X_1, X_2) = d_H(X_1, X_2)$.

Let $d_H \times d_H : \Pi_\mathcal{A} \times \Pi_\mathcal{A} \to \mathbf{R}^+$ be such that

$$d_H \times d_H((K, K'), (L, L')) = d_H(K, L) + d_H(K', L').$$

Then,

$$\begin{aligned} d_H \times d_H((K, K'), (M, M')) &= d_H(K, M) + d_H(K', M') \\ &\leq d_H(K, L) + d_H(L, M) + d_H(K', L') + d_H(L', M') \\ &= d_H \times d_H((K, K'), (L, L')) \\ &\quad + d_H \times d_H((L, L'), (M, M')). \end{aligned}$$

We have shown that $d_H \times d_H$ is a metric on $\Pi_\mathcal{A} \times \Pi_\mathcal{A}$.

Proposition 4. *$\cup, \cap : \Pi_\mathcal{A} \times \Pi_\mathcal{A} \to \Pi_\mathcal{A}$ are continuous maps with respect to $d_H \times d_H$ and d_H.*

Proposition 5. *$\neg : \Pi_\mathcal{A} \to \Pi_\mathcal{A}$ is continuous with respect to d_H and d_H.*

Let $D \times D : 2^U \times 2^U \to \mathbf{R}^+$ be such that

$$D \times D(X, Y) = D(X) + D(Y).$$

Then, it is a metric on $2^U \times 2^U$. Furthermore,

Proposition 6. *$\cup, \cap : 2^U \times 2^U \to 2^U$ are continuous maps with respect to $D \times D$ and D.*

Proposition 7. *$\neg : 2^U \to 2^U$ is continuous with respect to D and D.*

6 Metric and Measure on PFS

In the sequel, the technique of inducing metrics over the subsets of infinitary information systems is applied to induce a metric on the universe of a graded PFS. There is an natural way of inducing a graded indiscernibility relation in the context of a PFS. The above construction of the canonical PFS M_{max} induces a graded indiscernibility relation over the universe of Frege sets. This and the metric it induces on the universe of M_{max} can be captured by the following elaboration of the definition of a PFS.

Let $\mathcal{F} = (U, E, \ulcorner\cdot\urcorner, \llcorner\cdot\lrcorner)$ be a PFS and assume that E is the graded indiscernibility relation over U induced by a descending chain E_n, $n \in \omega$. One now has a metric d_H over $\mathit{\Pi}_{\mathcal{A}}$. The retraction mapping $\ulcorner\cdot\urcorner$ maps $\mathit{\Pi}_{\mathcal{A}}$–exact sets onto U and hence induces a metric d_h over U such that

$$d_h(\ulcorner Q\urcorner, \ulcorner P\urcorner) = d_H(Q, P).$$

Naturally, 2^U is equipped with the metric D. As observed earlier, d_H is the restriction of D to $\mathit{\Pi}_{\mathcal{A}}$–exact sets. Then $\mathcal{F}$ is *graded by* $E_n, n \in \omega$ and called a *graded PFS.*

Note that:

1. $\ulcorner\cdot\urcorner$ is continuous with respect to D and d_h,
2. $\llcorner\cdot\lrcorner$ is continuous with respect to d_h and D.

Assume $\mathcal{F}$ is graded by E_n, $n \in \omega$. It follows that

Proposition 8. *If $X \equiv Y$, then $d_H(\llcorner X\lrcorner, \llcorner Y\lrcorner) = 0$.*

Given an element X of 2^U, we say it is an *infinitesimal* if

$$d_H(X, \varnothing) = 0.$$

It immediately follows that for any $X, Y \in 2^U$,

$$d_H(X - Y, \varnothing) = 0 \Leftrightarrow d_H(X, Y) = 0.$$

We define a norm on every element X of 2^U as follows,

$$\|X\| = d_H(X, \varnothing).$$

It can readily be shown that $\|X\|$ defines a measure on 2^U.

7 Inversely Graded Indiscernibility and Metric

Given a set U, consider an ascending chain of equivalence relations $\{E_i\}$ over U such that $E_i \subseteq E_{i+1}$. We call the equivalence relation $E_\infty = \bigcup_{i\in\omega} E_i$ an *indiscernibility relation inversely graded by* $\{E_i\}$. By $[x]_i$ we mean the equivalence class $\{y : xE_iy\}$. It is clear that

$$[x]_n \subseteq [x]_{n+1}.$$

We have

$$[x]_{n+1} = \left[\bigcup\{[z]_n : z \in [x]_{n+1} - [x]_n\}\right] \cup [x]_n.$$

So, all (basic) open sets in Π_{n+1} are open in Π_n. This is to say

$$\Pi_{n+1} \subseteq \Pi_n.$$

Similarly, all (basic) open sets in Π_∞ are open in Π_n, i.e.,

$$\Pi_\infty \subseteq \Pi_n$$

for all $n \in \omega$.

Note that as the Pawlak topology is quasi–discrete, the collection of the *cogranules*, i.e.,

$$\mathcal{C}_n = \{U - [x]_n : x \in U\},$$

is a *cobase* base for a topology on U. Let

$$\mathcal{C}_\mathcal{R} = \bigcap_{n\in\omega} \mathcal{C}_n.$$

It can readily be shown that $\mathcal{C}_\mathcal{R}$ forms a cobase for a topology $\Pi_\mathcal{R}$ over U.

As the Pawlak topology is quasi–discrete and is a complete Boolean algebra, in lieu of duality, we have

Proposition 9. $\Pi_\mathcal{R} = \bigcap_{n\in\omega} \Pi_n = \Pi_\infty$.

This allows us to introduce the following definition: $T \subseteq U$ is *$\Pi_\mathcal{R}$–exact* iff it is Π_∞–exact, i.e.,

$$T = \cup\{[x]_\infty : x \in T\}.$$

It immediately follows that T is $\Pi_\mathcal{R}$–exact iff

$$T = Cl_{\Pi_\mathcal{R}}(T) = Int_{\Pi_\mathcal{R}}(T),$$

where $Cl_{\Pi_\mathcal{R}}(T)$ and $Int_{\Pi_\mathcal{R}}(T)$ are the closure and the interior of T with respect to the topology $\Pi_\mathcal{R}$. A subset of U which is not $\Pi_\mathcal{R}$-exact is called *$\Pi_\mathcal{R}$–rough.*

For each $n \in \omega$, we define a function $d_n : U \times U \to \mathbf{R}^+$, where $\mathbf{R}^+$ is the collection of nonnegative real numbers, as follows:

$$d_n(x.y) = \begin{cases} 0 \text{ if } [x]_n = [y]_n \\ 1 \text{ otherwise.} \end{cases}$$

Then

$$d_n(x,x) = 0, \quad d_n(x,y) = d_n(y,x).$$

Now define a function $d : U \times U \to \mathbf{R}^+$ as

$$d(x,y) = \sum_n 10^{-n} \cdot d_n(x,y),$$

where the convergence of the right side of the equation can readily be verified. It immediately follows that d is a pseudo-metric on U. (So is d_n.) This pseudo-metric d contains the information determining when x and y became discernible in the chain E_i. In particular,

Lemma 17. *$d(x,y) = 0.\overline{1} = \frac{1}{9}$ whenever $\neg(xE_\infty y)$.*

Recall that, since $(\Pi_{\mathcal{R}}, U, \varnothing, \cup, \cap, \neg)$ is a complete Boolean algebra, all open sets in $\Pi_{\mathcal{R}}$ are closed and vice versa. According to Hausdorff's account, the following defines a metric on $\Pi_{\mathcal{R}}$,

$$d_H(K,L) = \max\{\max_{x\in K} dist(x,L), \max_{x\in L} dist(x,K)\}$$

where

$$dist(x,L) = \min\{d(x,z) : z \in L\}.$$

It is well–known that every $\Pi_{\mathcal{R}}$–rough set X can be characterized by a pair of $\Pi_{\mathcal{R}}$–exact sets (Q,T) where

$$Q = Cl_{\Pi_{\mathcal{R}}}(X), \quad T = U - Int_{\Pi_{\mathcal{R}}}(X).$$

Given two $\Pi_{\mathcal{R}}$-rough sets X_1, X_2, define the distance metric D by

$$D(X_1,X_2) = \max\{d_H(Q_1,Q_2), d_H(T_1,T_2)\}.$$

If X is $\Pi_{\mathcal{R}}$–exact then,

$$Q = Cl_{\Pi_{\mathcal{R}}}(X) = X, \quad T = U - Int_{\Pi_{\mathcal{R}}}(X) = U - X.$$

It can readily be verified that

$$d_H(Q_1,Q_2) = d_H(X_1,X_2) = d_H(T_1,T_2).$$

Therefore, $D(X_1,X_2) = d_H(X_1,X_2)$.

Let $d_H \times d_H : \Pi_{\mathcal{R}} \times \Pi_{\mathcal{R}} \to \mathbf{R}^+$ be such that

$$d_H \times d_H((K,K'),(L,L')) = d_H(K,L) + d_H(K',L').$$

Then,

$$\begin{aligned} d_H \times d_H((K,K'),(M,M')) &= d_H(K,M) + d_H(K',M') \\ &\le d_H(K,L) + d_H(L,M) + d_H(K',L') + d_H(L',M') \\ &= d_H \times d_H((K,K'),(L,L')) \\ &\quad + d_H \times d_H((L,L'),(M,M')). \end{aligned}$$

We have shown that $d_H \times d_H$ is a metric on $\Pi_{\mathcal{R}} \times \Pi_{\mathcal{R}}$.

Proposition 10. *$\cup, \cap : \Pi_{\mathcal{R}} \times \Pi_{\mathcal{R}} \to \Pi_{\mathcal{R}}$ are continuous maps with respect to $d_H \times d_H$ and d_H.*

Proposition 11. *$\neg : \Pi_{\mathcal{R}} \to \Pi_{\mathcal{R}}$ is continuous with respect to d_H and d_H.*

Let $D \times D : 2^U \times 2^U \to \mathbf{R}^+$ be such that

$$D \times D(X,Y) = D(X) + D(Y).$$

Then, $D \times D$ is a metric on $2^U \times 2^U$. Furthermore,

Proposition 12. *$\cup, \cap : 2^U \times 2^U \to 2^U$ are continuous maps with respect to $D \times D$ and D.*

Proposition 13. *$\neg : 2^U \to 2^U$ is continuous with respect to D and D.*

8 An Example of Inversely Graded Indiscernibility

Let $\mathcal{L}_{\mathcal{R}}$ be the first order language of real numbers. Let $\mathcal{R}$ be the standard model of $\mathcal{L}_{\mathcal{R}}$ and let R be the universe of $\mathcal{R}$. $X \subseteq R$ is said to be *definable* iff there is a formula $A(x)$ of $\mathcal{L}_{\mathcal{R}}$ such that

$$X = \{x \in \mathcal{R} \mid \mathcal{R} \vDash A(x)\}.$$

Let $X \subseteq \mathbf{M}$ and $\mathcal{D}(X)$ be the family of all definable subsets of X. $\mathcal{F} \subseteq \mathcal{D}(X)$ is a *d-filter* on X iff

$$\begin{array}{l} X \in \mathcal{F} \\ A, B \in \mathcal{F} \Rightarrow (A \cap B) \in \mathcal{F} \\ A \in \mathcal{F}, A \subseteq B \in \mathcal{D}(X) \Rightarrow B \in \mathcal{F} \\ \emptyset \notin \mathcal{F}. \end{array}$$

It is clear that if $\mathcal{F}$ is a d-filter on X then,

$$A \in \mathcal{F} \Rightarrow (X - A) \notin \mathcal{F}.$$

Let $\mathcal{F}$ be a d-filter on X. If

$$A \in 2^X \Rightarrow A \in \mathcal{F} \vee (X - A) \in \mathcal{F}$$

then $\mathcal{F}$ is said to be an *ultra d–filter on* $\mathcal{D}(X)$.

Let $\mathcal{F} \subseteq \mathcal{D}(X)$; $\mathcal{F}$ is *d–consistent* iff the closure of $\mathcal{F}$ under finite intersections does not contain $\emptyset$ as an element. Let

$$UC(\mathcal{F}) =_{df} \{Y \in \mathcal{D}(X) \mid (\exists F \in \mathcal{F})(F \subseteq Y)\}.$$

Lemma 18. *Let $\mathcal{F}$ be a d–filter on X. Let $Y \subseteq X$. Suppose $\{Y\} \cup \mathcal{F}$ is not d–consistent. Then $\{X - Y\} \cup \mathcal{F}$ is d–consistent.*

Since $\mathcal{D}(X)$ is countably infinite, it may be enumerated as $\mathcal{D}(X)_i : i < \omega$.

Lemma 19. *(***Lindenbaum***) Every d–filter $\mathcal{F}$ has an ultra d–filter extension.*

Proof. Define an infinite sequence,

$$\Gamma_0 \subseteq \Gamma_1 \subseteq \ldots \subseteq \Gamma_k \subseteq \ldots \subseteq \Gamma_\omega,$$

of families of definable subsets of X by

$$\begin{array}{rll} \Gamma_0 = & \mathcal{F}, & \\ \Gamma_{k+1} = & UC(\Gamma_k \cup \{\mathcal{D}(X)_{k+1}\}) & \text{if this } \Gamma_k \cup \{\mathcal{D}(X)_{k+1}\} \text{ is d-consistent} \\ & \Gamma_k & \text{otherwise,} \end{array}$$

and

$$\Gamma_\infty = \cup\{\Gamma_k \mid k < \omega\}.$$

Then, it follows that $\varGamma_\omega$ is a maximal d–filter which contains the d–filter $\mathcal{F}$. Furthermore, $\varGamma_k,\ k < \omega$ are all filters. Let $\{S_i \mid i \in \mathcal{D}(X)\}$ be a $\mathcal{D}(X)$-family of sets. On the product,

$$\varPi_{i \in I} S_i = \{g : \mathcal{D}(X) \to \cup_{i \in I} S_i \mid \text{for all } i \in \omega,\ g(i) \in S_i\},$$

we define the following equivalence relation,

$$g_1 \equiv_{\varGamma_k} g_2 \Leftrightarrow \{i \in \mathcal{D}(X) \mid g_1(i) = g_2(i)\} \in \varGamma_k,\ k \in \omega.$$

It is clear that $\equiv_{\varGamma_k},\ k \in \omega$ is a descending ω–sequence of equivalence relations and,

$$g_1 \equiv_{\varGamma_\infty} g_2 \Leftrightarrow \{i \in \mathcal{D}(X) \mid g_1(i) = g_2(i)\} \in \varGamma_\infty,$$

is the limit of it.

Then each cofinite set is definable (even in first order Peano arithmetic). Therefore the collection of cofinite sets $\mathcal{F}$ is a d–filter. In lieu of Lindenbaum's Lemma, there is an ω–chain,

$$\mathcal{F} = \varGamma_0 \subseteq \varGamma_1 \subseteq \ldots \subseteq \varGamma_k \subseteq \ldots \subseteq \varGamma_\infty,$$

of filters approximating an ultra–filter $\varGamma_\infty$. Define an ultra–power $\mathcal{R}^*$ of $\mathcal{R}$ by

$$\mathcal{R}^* = \mathcal{R}^\omega / \mathcal{F}.$$

$\varGamma_\infty$ yields an indiscernibility relation $\equiv_{\varGamma_\infty}$on $\mathcal{R}^*$ inversely graded by $\equiv_{\varGamma_k},\ k \in \omega$. This in turn induces a metric on $\mathcal{R}^*$ as discussed in the previous section.

9 Metrically Open Subsets of a Graded Plenum

We conclude our study of exact subsets of infinite information systems with an open problem in the theory of the graded plenum M_{max}.

Let $(U, E, \ulcorner\cdot\urcorner, \llcorner\cdot\lrcorner)$ be a plenum graded by $E_n,\ n \in \omega$. The metric $d(x, y) : U \times U \to R$ induced by the graded indiscernibility relation on U gives rise to the metric space (U, d). The collection of all d–open balls forms a basis for a topology (U, τ_d) called the *metric topology* of (U, d).

By effectivizing the SFP construction of D_∞ as in effective domain theory [30], we can single out the computable elements of D_∞ and thence define "computable" elements of M_{max} as those whose maximal continuous approximation are computable. As effective DT indicates, these computable elements are now represented by Gődel numbers (programs) and we can define "computable exact subset of M_{max}" (equivalently, "computable element of M_{max}") by effectively enumerating these programs. The recursion theorist A.I. Mal'cev [40] proposed two different versions of the notion of a "computable program set" in the context of numeration theory, the general theory of Gődel numberings [17, 18]. The one of interest to RST deals with "index sets" − sets of natural numbers which are closed under the relation of program equivalence − and "extensional" program transformations – effective number theoretic functions which preserve program equivalence. Accordingly, this extensional recursion theory identifies the computable program sets with the r.e. index sets, i.e., r.e. exact sets.

Given an acceptable indexing of r.e. subsets of natural numbers, Myhill–Sheperdson's [41] result in recursion theory states that every extensional program transformation is an effective enumeration operator. (As a corollary, we have the result by Rice–Shapiro [53] that every r.e. index set is upward closed under subset inclusion). This and Thomas Streicher [61], and Yu Ershov's [19] result in effective DT established that extentional computability entails continuity.

It thus appears natural to conjecture that all computable elements of M_{max} are metric open.

As RST passes from finite (definitely computable) to infinite information systems it may be desirable to consider the first steps, towards computably infinite information systems.

References

1. Aczel P. [1980]: "Frege Structure and the Notions of Proposition, Truth and Set". In Eds. J. Barwise, H. Keisler and K. Kunen, *The Kleene Symposium*, North-Holland. pp. 31-59.
2. Apostoli P. and A. Kanda [2000]: "Approximation Spaces of Type-Free Sets," *Proc. of RSCTC'00*, Eds. W. Ziarko, Y. Y. Yao. Lecture Notes in Artificial Intelligence (LNAI) **2005**. Eds. J. G. Carbonell, J. Siekmann, Springer-Verlag. pp. 98-105.
3. Apostoli P., Kanda, A. *Parts of the Continuum: towards a modern ontology of science*, forthcoming in Poznan Studies in the Philosophy of Science and the Humanities, Ed. Leszek Nowak.
4. Apostoli P., Kanda, A. "Rough Sets and the Continuum." Submitted to *Advances in Rough Sets J.*, Eds-in-Chief J. Peters and A. Skowron. Springer LNAI subseries.
5. Bell J. L. [1995]: "Type Reducing Correspondences and Well-Orderings: Frege's and Zermelo's Constructions Re-examined," *J. of Symbolic Logic*, **60**.
6. Bell J. L. [2000]: "Set and Classes as Many," *J. of Philosophical Logic*, **29** (6), pp. 595-681.
7. Blamey S. [1986]: "Partial Logic," *Handbook of Philosophical Logic*, Eds. Gabbay D. and Guenthner F., D. Reidel Publishing Company, **III**, pp. 1-70.
8. Boolos G. [1987b]: "On the Consistency of Frege's *Foundations of Arithmetic*," In Ed. J. L. Thomson, *On Being and Saying: Essays for Richard Cartwright*, pp. 3-20. Cambridge, Mass., The MIT Press. Reprinted in [10, Boolos 1998].
9. Boolos G. [1990b]: "The Standard of Equality of Numbers," in Ed. Boolos G., *Meaning and Method: Essays in Honor of Hilary Putnam*, Cambridge: Cambridge University Press. Reprinted in [10, Boolos 1998].
10. Boolos G. [1998]: *Logic, Logic, and Logic*, Ed. R. C. Jeffrey. Cambridge, Mass.: Harvard University Press.
11. Cantor G. [1883]: "Uber unendliche, lineare Punktmannigfaltigkeiten, 5", *Mathematische Annalen* **21**, pp. 545-86. In [13, pp. 165-209]. A seperate edition of this, with a preface, was printed under the title *Grundlagen einer allegmeinen Mannigfaltigkeitslehre. Ein mathematisch-philosophischer Versuch in der Lehre des Undendliche.* Tuebner, Leipzig.

12. Cantor G. [1895]: “Beitraĝe zur Beründung der transfiniten Mengenehre, I, II” *Mathematische Annalen* **46**, pp. 481-512; **49**, pp. 207-46. In [13, pp. 282-56].
13. Cantor G. [1962]: *Gesammelte Abhandlungen Mathenatischen und Philosopischen Inhalts*. Ed. by E. Zermelo. Berlin: Springer; reprinted, Hildesheim: Olms.
14. Cattaneo G. [1998]: “Abstract Approximation Spaces for Rough Theories”, *Rough Sets in Knowledge Discovery: Methodology and Applications*, Eds. L. Polkowski, A. Skowron. Studies in Fuzziness and Soft Computing. Ed. J. Kacprzyk. **18** Springer.
15. Cocchiarella N. [1972a]: “Whither Russell’s Paradox of Predication?”, *Logic and Ontology*, **2** of *Studies in Contemporary Philosophy*, Ed. K. Muntz, New York University Press.
16. Cocchiarella N. [1972b]: “Properties as Individuals in Formal Ontology”, $NO\widehat{U}S$, **VI**, No. 2, May, pp. 165-188.
17. Ershov J. L. [1973]: “Theorie der Numerierungen I”, *Zeitschrift f. Math. und Grundlagen d. Mathematik*, **19** pp. 289-388.
18. Ershov J. L. [1975]: “Theorie der Numerierungen II”, *Zeitschrift f. Math. und Grundlagen d. Mathematik*, **21** pp. 473-584.
19. Ershov J. L. [1977]: “Model C of Partial Continuous Functionals”, *Proc. Logic Colloquium*, (1976). North Holland, Amsterdam, pp. 455-467.
20. Feferman S. [1984]: “Towards Useful Type-free Theories I”. *Journal of symbolic Logic*, **49**, pp. 75-111.
21. Frege G. [1884]: *Die Grundlagen der Arithmetik. Eine logisch mathematische Untersachung uber den Begridd der Zahl.* Breslau: William Koebner. English translation by J. L. Austin, *The Foundations of Arihmetic*. Oxford, Basil Blackwell (1950).
22. Frege G. [1893, 1903]: *Grundgesetze der Arithmetik,* **1**, **2**, Jena: Verlag Hermann Pohle. Reprinted at Hildesheim. Volume **1** is partially translated in [26]. Vol. **2** is translated in part in [27].
23. Frege G. [1902]: “Letter to Russell”, in [62, pp. 126-128].
24. Frege G. [1903a]: “The Russell Paradox”, in Frege, Gottlob, [27], pp. 127-143. Abridged and reprinted in Irvine, A.D., *Bertrand Russell: Critical Assessments*, vol. 2, London: Routledge, (1999), 1-3.
25. Frege G. [1903b]: “Concerning Zermelo’s Independent Discovery of Russell’s Paradox”, in *Frege Against the Formalists.*
26. Frege G. [1960]: *Translations from the Philosophical Writings of Gottlob Frege.* Edited and translated by Peter Geach and Max Black. Oxford: Basil Blackwell.
27. Furth M. [1964]: *The Basic Laws of Arithmetic: Expositition of the system*, Berkeley: University of California Press.
28. Gilmore P.C. [1986]: “Natural Deduction Based Set Theories: A New Resolution of the Old Paradoxes”, *J. of Symbolic Logic*, **51**, pp. 394-411.
29. Kanda A. [1979]: “When are Two Effectively given Domains Equal?”, *Proc. of the 4th GI Theoretical Computer Science Symposium*, Lecture Notes in Computer Science (LNCS) textbf67 (with D. Park).
30. Kanda A. [1979]: “Fully Effective Solutions of Recursive Domain Equations”, *Proc. of the 8th International Symposium on Mathematical Foundations of Computer Science*, LNCS **74**.
31. Kanda A. [1981]: “Constructive Category Theory”, *Proc. of the 10th International Symposium on Mathematical Foundations of Computer Science*, LNCS **118**.
32. Kanda A. [1988]: “Recursion Theorems and Effective Domains”, *Annals of Pure and Applied Logic*, **38**, pp. 289-300.
33. Kleene S. C. [1952]: *Introduction to Metamathematics*, North-Holland Publishing Co., Amsterdam.

34. Kleene S. C. and Rosser J. S. [1935]: " The Inconsistency of Certain Formal Logics", *Annals of Mathematics*, **36**, pp. 630-636.
35. Kripke S. [1963]: "Semantical Analysis of Modal Logic I. Normal Modal Propositional Calculi". *Zeitschrift f. Math. Logik und Grundlagen d. Math.*, **9**.
36. Leśniewski S. [1992]: "Foundations of the General Theory of Sets", in Eds. Surma Srzednicki and Barnett Rickey, *Stanislaw Leśniewsky Collected Works.* Kluwer Academic Publishers Dordrecht, pp. 128-173.
37. Lewis D. [1979]: "Counterpart Theory and Quantified Modal Logic", *J. of Philosophy* **65** (1968), pp. 113-26. Reprinted in Michael J. Loux, Ed., *The Possible and the Actual* (Cornell U.P.).
38. Lewis D. K. [1986]: *On the Plurality of Worlds.* Oxford: Basil Blackwell.
39. Lewis D. K. [1991]: *Parts of Classes.* Oxford: Basil Blackwell.
40. Mal'cev A. I. [1971]: *The Mathematics of Algebraic Systems*, North-Holland, Amsterdam.
41. Myhill J., Shepardson J.C. [1955]: "Efective operations on partial recursive functions", *Zeitschrift f. Math. Logik und Grundlagen d. Mathematik* textbf1 pp. 310-317.
42. Orlowska E. [1985]: "Semantics of Vague Concepts", in *Foundations of Logic and Linguistics, Problems and Their Solutions*, Eds. G. Dorn and P. Weingartner, Plenum Press, NewYork.
43. Parsons C. [1983]: "Frege's Theory of Number", in Parsons C., *Mathematics in Philosophy: Selected Essays.* Ithaca: Cornell University Press.
44. Pawlak Z. [1982]: "Rough Sets, Algebraic and Topological Approaches", *International Journal of Computer and Information Sciences*, **11**, pp. 341-366.
45. Pawlak Z. [1991]: *Rough Sets: Theoretical aspects of reasoning about data.* Kluwer, Dordrecht.
46. Pawlak Z. [1998]: "Rough Set Elements", *Rough Sets in Knowledge Discovery: Methodology and Applications*, Eds. L. Polkowski, A. Skowron. Studies in Fuzziness and Soft Computing. Ed.: J. Kacprzyk. **18**, Springer.
47. Plotkin G. [1976]: "A Power Domain Construction", *SIAM Journal on Computing*, **5**, pp. 452-487.
48. L. Polkowski, A. Skowron [1996]: "Rough mereology: A new paradigm for approximate reasoning", *Journal of Approximate Reasoning* **15**(4), pp. 333-365.
49. L. Polkowski, A. Skowron [1999]: "Towards adaptive calculus of granules", In L.A. Zadeh and J. Kacprzyk Eds., *Computing with Words in Information/Intelligent Systems 1: Foundations*, Springer- Verlag Group (Physica-Verlag), Studies in Fuzziness and Soft Computing **30**, pp. 201-228.
50. L. Polkowski, A. Skowron [2000]: "Rough-neuro computing", *Proc. of RSCTC'00*, Eds. W. Ziarko, Y. Y. Yao. LNAI **2005**.
51. Polkowski L. [2002]: "On Asymptotic Properties of Rough-Set-Theoretic Approximations. Fractal Dimension, Exact Sets, and Rough Inclusion in Potentially Infinite Information Systems", RSCTC 2002. LNAI **2475**.
52. Polkowski L. [2002]: "On Fractal Dimension in Information Systems: Towards exact sets in infinite information systems", *Fundamenta Informaticae* **50**, pp. 3-4.
53. Rice, H. G. [1956]: "On Completely recursively enumerable classes and their key arrays", *J. of Symbolic Logic* **21**, pp. 304-308.
54. Robinson A. [1970]: *Non-Standard Analysis*, North-Holland, Amsterdam.
55. Russell B.A.W. [1902]: "Letter to Frege", in [62, pp. 124-125].
56. Scott D. [1971]: "Continuous lattices", *Lecture Notes in Math.*, **274**, Springer, Berlin (1972). pp. 97-136.

57. Scott D. [1975]: “Combinators and Classes”, LNCS, **37**, pp. 1-26.
58. Scott D. [1976]: “Data Types as Lattices,” *SIAM Journal on Computing*, **5**, pp. 522-587.
59. Smyth M. [1978]: “Power Domains”, *Journal of Computer and Systems Science*, **16**, pp. 23-36.
60. Smyth M. and G. Plotkin [1977]: “The Categorical Solutions of Recursive Domain Equations”, *Proc. of the 18th FOCS Conference*.
61. Streicher, T. [1982]: *Scott Domains, Computability and the Definability Problem*, Diplomarbeit, Johanenes Kepler Universitat, Lintz. CAMP Report 82-7.
62. van Heijenoort J. [1967]: *From Frege to Godel: A Source Book in Mathematical Logic, 1879 - 1931*. Cambridge Mass.: Harvard University Press.
63. Wiweger A. [1989]: “On Topological Rough Sets”, *Bulletin of the Polish Academy of Sciences*, Mathematics, **37**, No. 16.
64. Wright C. [1983]: *Frege’s Conception of Numbers as Objects*. Scots Philosophical Monographs, textbf2, Aberdeen, Aberdeen University Press.
65. Yao Y. Y. [1998a]: “Constructive and Algebraic Methods of the Theory of Rough Sets”, *Journal of Information Sciences* **109**, pp. 21-47.
66. Yao Y. Y. [1998b]: “On Generalizing Pawlak Approximation Operators”, Eds. L. Polkowski and A. Skowron, *RSCTC’98*, LNAI **1414**, Eds. J. G. Carbonell, J. Siekmann, Springer-Verlag, pp. 289-307.

Data Structure and Operations for Fuzzy Multisets

Sadaaki Miyamoto

Institute of Engineering Mechanics and Systems, University of Tsukuba,
Ibaraki 305-8573, Japan

Abstract. An overview of data structures and operations for fuzzy multisets is given. A simple linear list identified with an infinite dimensional vector is taken as an elementary data structure for fuzzy multisets. Fuzzy multiset operations are defined by corresponding vector operations. Two level sets of α-cut and ν-cut are defined and commutativity between them are described. Spaces of fuzzy multisets are also discussed.

1 Introduction

Multisets which have been considered by several authors (e.g., [6, 9, 1]) with application to database queries (e.g., [3]) are attracting more attention of researchers in relation to new computation paradigms [2].

Fuzzy multisets have also been studied [23, 5, 10, 19, 18, 11, 7]. This paper is not intended to provide a comprehensive overview, but focuses several features of fuzzy multisets with discussion of data structure for representing fuzzy multisets. A linear list or a simple vector is taken as the elementary data structure. This structure is not only for convenience but reveals fundamental characteristic of multisets as results of linear data processing. Thus a novel type of image of a function is discussed.

A characteristic in fuzzy multisets is their inherent infiniteness. We hence discuss infinite–dimensional spaces of fuzzy multisets.

We omit most of proofs of propositions, as they are not difficult and readers can try to prove by themselves. Moreover applications are not discussed here, they are studied in other papers [12, 13, 16].

2 Preliminaries

Assume throughout this paper that the universal set X is finite. We thus write $X = \{x_1, x_2, \dots, x_n\}$ or $X = \{x, y, \dots, z\}$. Scalars are denoted by $a, b, \dots$

Fuzzy Sets

To show operations for fuzzy multisets as extensions of those for fuzzy sets, we first provide notations for fuzzy sets. Infix notations for max and min are denoted by $\vee$ and $\wedge$, respectively as in most literature [25].

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 189–200, 2004.

As is well–known, for fuzzy sets A and B of X:

1. $$A \subseteq B \iff \mu_A(x) \le \mu_B(x), \quad \forall x \in X.$$
2. $$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x).$$
3. $$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x).$$
4. A weak α-cut is denoted by $[A]_\alpha$ here:
   $$x \in [A]_\alpha \iff \mu_A(x) \ge \alpha.$$
   Moreover, a strong α-cut is denoted by $]A[_\alpha$:
   $$x \in]A[_\alpha \iff \mu_A(x) > \alpha.$$
5. The t-norms and conorms are used as generalizations of fuzzy set union and intersection. We omit details of t-norms [20, 4], and note only that they are respectively denoted by $t(a,b)$ and $s(a,b)$. When used as set operations, the operations are denoted by
   $$\mu_{A\mathbf{T}B}(x) = t(\mu_A(x), \mu_B(x)), \quad \mu_{A\mathbf{S}B}(x) = s(\mu_A(x), \mu_B(x)).$$
   A particular t-norm of algebraic product is denoted by $a \cdot b$ whereby
   $$\mu_{A\cdot B}(x) = \mu_A(x) \cdot \mu_B(x),$$

Important properties of commutativity between an operation and an α-cut hold:

$$\begin{aligned}
[A \cup B]_\alpha &= [A]_\alpha \cup [B]_\alpha, \\
[A \cap B]_\alpha &= [A]_\alpha \cap [B]_\alpha, \\
]A \cup B[_\alpha &=]A[_\alpha \cup]B[_\alpha, \\
]A \cap B[_\alpha &=]A[_\alpha \cap]B[_\alpha.
\end{aligned}$$

We also have

$$\begin{aligned}
A \subseteq B &\iff [A]_\alpha \subseteq [B]_\alpha, \quad \forall \alpha \in (0,1], \\
A \subseteq B &\iff]A[_\alpha \subseteq]B[_\alpha, \quad \forall \alpha \in [0,1).
\end{aligned}$$

For a given function $f\colon X \to Y$, we moreover have

$$]f(A)[_\alpha = f(]A[_\alpha).$$

Multisets

A multiset M of X is characterized by the count function $C_M \colon X \to \boldsymbol{N}$, where $\boldsymbol{N} = \{0, 1, 2, \ldots\}$. Thus, $C_M(x)$ is the number of occurrences of the element $x \in X$.

The following is basic relations and operations for crisp multisets;

I. (inclusion):

$$M \subseteq N \Leftrightarrow C_M(x) \leq C_N(x), \ \ \forall x \in X.$$

II. (equality):

$$M = N \Leftrightarrow C_M(x) = C_N(x), \ \ \forall x \in X.$$

III. (union):

$$C_{M \cup N}(x) = C_M(x) \vee C_N(x).$$

IV. (intersection):

$$C_{M \cap N}(x) = C_M(x) \wedge C_N(x).$$

V. (sum):

$$C_{M+N}(x) = C_M(x) + C_N(x).$$

VI. (Cartesian product): For $x \in X$ and $y \in Y$,

$$C_{M \times N}(x, y) = C_M(x) \cdot C_N(y).$$

VII. ($\boldsymbol{\nu}$-cut):

A ν-cut $M \mapsto [M]^\nu$ transforms a multiset into an ordinary set:

$$C_M(x) \geq \nu \Leftrightarrow x \in [M]^\nu$$
$$C_M(x) < \nu \Leftrightarrow x \notin [M]^\nu$$

It is reasonable to assume that the number $C_M(\cdot)$ should be finite.

Example 1. Let us consider a simple example of $X = \{a, b, c, d\}$, $C_M(a) = 2$, $C_M(b) = 3$, $C_M(c) = 1$, and $C_M(d) = 0$. We write $M = \{2/a, 3/b, 1/c\}$ by ignoring zero occurrence $0/d$. Alternatively, we can write $M = \{a, a, b, b, b, c\}$. notice also that we can exchange the order of elements in $\{\cdots\}$ as in ordinary sets

$$\{a, a, b, b, b, c\} = \{c, a, b, b, a, b\}. \tag{1}$$

A Sorting Operation. A sorting operation is essential in processing multisets. Let us take the last example in which $\{c, a, b, b, a, b\}$ should be arranged into the sorted form of $\{a, a, b, b, b, c\}$ in order to check (1). This means that we actually handle sequences (c, a, b, b, a, b) and (a, a, a, b, b, c) as an abstract data structure instead of $\{a, a, b, b, b, c\}$ and $\{c, a, b, b, a, b\}$.

3 Fuzzy Multisets

Fuzzy multiset A of X (often called fuzzy bag) is characterized by the function $C_A(\cdot)$ of the same symbol, but the value $C_A(x)$ is a finite multiset in $I = [0,1]$ [23]. In other words, given $x \in X$,

$$C_A(x) = \{\mu, \mu', \dots, \mu''\}, \qquad \mu, \mu', \dots, \mu'' \in I.$$

Assume $\mu, \mu', \dots, \mu''$ are nonzero for simplicity. We write

$$A = \{\{\mu, \mu', \dots, \mu''\}/x, \dots\}$$

or

$$A = \{(x, \mu), (x, \mu'), \dots, (x, \mu''), \dots\}.$$

As a data structure, we introduce an infinite–dimensional vector:

$$C_A(x) = (\mu, \mu', \dots, \mu'', 0, 0, \dots).$$

Collection of such vectors is denoted by $\mathcal{V}$:

$$\mathcal{V} = \{\, (\mu, \mu', \dots, \mu'', 0, 0, \dots) : \quad \mu, \mu', \dots, \mu'' \in I \,\}$$

A sorting operation to multisets in I is important in defining operations for fuzzy multisets. This operation denoted by S ($S\colon \mathcal{V} \to \mathcal{V}$) rearranges the sequence in $\mathcal{V}$ into the decreasing order:

$$S((\mu, \mu', \dots, \mu'', 0, 0, \dots)) = (\nu^1, \nu^2, \dots, \nu^p, 0, 0, 0)$$

where

$$\nu^1 \geq \nu^2 \geq \dots \geq \nu^p > 0$$

and

$$\{\mu, \mu', \dots, \mu''\} = \{\nu^1, \nu^2, \dots, \nu^p\}.$$

Thus we can assume

$$C_A(x) = (\nu^1, \nu^2, \dots, \nu^p, 0, 0, 0) \tag{2}$$

Moreover we define the length of $C_A(x)$ by

$$L(C_A(x)) = p$$

The above sorted sequence for $C_A(x)$ is called the standard form for a fuzzy multiset, as many operations are defined in terms of the standard form.

Additional operations on $\mathcal{V}$ are necessary in order to define fuzzy multiset operations. Assume

$$k = (k_1, k_2, \dots, k_q, \dots, k_r, 0, 0, \dots),\ l = (l_1, l_2, \dots, l_q, \dots, l_s, 0, 0, \dots) \in \mathcal{V}$$

where k_i $(i = 1, \dots, r)$ and l_j $(j = 1, \dots, s)$ are nonzero.

Then we define

$$
\begin{aligned}
k \vee l &= (\max\{k_1, l_1\}, \dots, \max\{k_q, l_q\}, \dots), && (3)\\
k \wedge l &= (\min\{k_1, l_1\}, \dots, \min\{k_q, l_q\}, \dots), && (4)\\
k \cdot l &= (k_1 \cdot l_1, \dots, k_q \cdot l_q, \dots), && (5)\\
k \mid l &= (k_1, k_2, \dots, k_r, l_1, l_2, \dots, l_s, 0, 0, \dots) && (6)
\end{aligned}
$$

Assume $k_1 \geq k_2 \geq \cdots \geq k_r$; let $\alpha \in (0,1]$ and $\nu \in \boldsymbol{N}$. Then we define

$$
\begin{aligned}
|k| &= \sum_{i=1}^{r} k_i, && (7)\\
[[k]]_\alpha &= j \quad (k_j \geq \alpha, k_{j+1} < \alpha), && (8)\\
]]k[[_\alpha &= j' \quad (k_{j'} > \alpha, k_{j'+1} \leq \alpha), && (9)\\
[[k]]^\nu &= k_\nu && (10)
\end{aligned}
$$

Note that in the last equation (10), $k_\nu = 0$ if and only if $\nu > r$.

Moreover we define inequality of the two vectors:

$$k \leq l \iff k_i \leq l_i, \quad i = 1, 2, \dots \tag{11}$$

We now define fuzzy multiset operations.

1. **inclusion:**

$$A \subseteq B \iff S(C_A(x)) \leq S(C_B(x)), \quad \forall x \in X.$$

2. **equality:**

$$A = B \iff S(C_A(x)) = S(C_B(x)), \quad \forall x \in X.$$

3. **union:**

$$C_{A \cup B}(x) = S(C_A(x)) \vee S(C_B(x)).$$

4. **intersection:**

$$C_{A \cap B}(x) = S(C_A(x)) \wedge S(C_B(x)).$$

5. **sum:**

$$C_{A+B}(x) = S(S(C_A(x)) \mid S(C_B(x))).$$

6. **product:**

$$C_{A \cdot B}(x) = S(C_A(x)) \cdot S(C_B(x)).$$

7. **$\boldsymbol{\alpha}$-cuts:**

$$
\begin{aligned}
C_{[A]_\alpha}(x) &= [[S(C_A(x))]]_\alpha,\\
C_{]A[_\alpha}(x) &=]]S(C_A(x))[[_\alpha.
\end{aligned}
$$

8. ***ν*-cut:**

$$C_{[A]^{\nu}}(x) = [[S(C_A(x))]]^{\nu}.$$

9. **Cardinality:**

$$|A| = \sum_{x \in X} |C_A(x)|.$$

We obviously have

$$\begin{aligned}
L(C_{A \cup B}(x)) &= \max\{L(C_A(x)), L(C_B(x))\}, \\
L(C_{A \cap B}(x)) &= \min\{L(C_A(x)), L(C_B(x))\}, \\
L(C_{A+B}(x)) &= L(C_A(x)) + L(C_B(x)), \\
L(C_{A \cdot B}(x)) &= \min\{L(C_A(x)), L(C_B(x))\}.
\end{aligned}$$

The reason why we define fuzzy multiset operations using the sorting operation S is shown in next propositions.

Proposition 1. An ordinary fuzzy set F and a crisp multiset M can be embedded into the collection of all fuzzy multisets. Let the embedding map be $\mathcal{I}$. Namely,

$$C_{\mathcal{I}(F)}(x) = (\mu_F(x), 0, 0, \dots),$$

and

$$C_{\mathcal{I}(M)}(x) = (1, 1, \dots, 1, 0, 0, \dots),$$

where the number of 1's in the right hand side is equal to $C_M(x)$. Assume F_1 and F_2 are ordinary fuzzy sets, and M_1 and M_2 are crisp multisets. We then have

$$\begin{aligned}
\mathcal{I}(F_1 \cup F_2) &= \mathcal{I}(F_1) \cup \mathcal{I}(F_2), \\
\mathcal{I}(F_1 \cap F_2) &= \mathcal{I}(F_1) \cap \mathcal{I}(F_2), \\
\mathcal{I}(M_1 \cup M_2) &= \mathcal{I}(M_1) \cup \mathcal{I}(M_2), \\
\mathcal{I}(M_1 \cap M_2) &= \mathcal{I}(M_1) \cap \mathcal{I}(M_2), \\
\mathcal{I}(M_1 + M_2) &= \mathcal{I}(M_1) + \mathcal{I}(M_2), \\
[\mathcal{I}(F)]_{\alpha} &= \mathcal{I}([F]_{\alpha}), \\
]\mathcal{I}(F)[_{\alpha} &= \mathcal{I}(]F[_{\alpha}), \\
[\mathcal{I}(M)]^{\nu} &= \mathcal{I}([M]^{\nu}).
\end{aligned}$$

Proposition 2. For two fuzzy multisets A and B of X, the following relations hold.

$$\begin{aligned}
&A \subseteq B \iff [A]_\alpha \subseteq [B]_\alpha, \quad \forall \alpha \in (0,1], \\
&A \subseteq B \iff]A[_\alpha \subseteq]B[_\alpha, \quad \forall \alpha \in [0,1). \\
&[A \cup B]_\alpha = [A]_\alpha \cup [B]_\alpha, \\
&[A \cap B]_\alpha = [A]_\alpha \cap [B]_\alpha, \\
&[A + B]_\alpha = [A]_\alpha + [B]_\alpha, \\
&]A \cup B[_\alpha =]A[_\alpha \cup]B[_\alpha, \\
&]A \cap B[_\alpha =]A[_\alpha \cap]B[_\alpha, \\
&]A + B[_\alpha =]A[_\alpha +]B[_\alpha, \\
&[A \cup B]^\nu = [A]^\nu \cup [B]^\nu, \\
&[A \cap B]^\nu = [A]^\nu \cap [B]^\nu.
\end{aligned}$$

The proof is easy and omitted. Notice also that

$$[A + B]^\nu \neq [A]^\nu + [B]^\nu$$

in general.

Proposition 3. For fuzzy multisets A, B, and C of X, the commutative, associative, and distributive laws holds for operations $\cup$ and $\cap$. Namely, the collection of all fuzzy multisets of X forms a distributive lattice.

$$\begin{aligned}
A \cup B &= B \cup A, \\
A \cap B &= B \cap A, \\
A \cup (B \cup C) &= (A \cup B) \cup C, \\
A \cap (B \cap C) &= (A \cap B) \cap C, \\
(A \cap B) \cup C &= (A \cup C) \cap (B \cup C), \\
(A \cup B) \cap C &= (A \cap C) \cup (B \cap C).
\end{aligned}$$

The proof is omitted here (see Chapter 2 of [7]). Readers should note the properties in Proposition 2 is used in the proof.

Proposition 4. Let A be an arbitrary fuzzy multiset of X. Then,

$$\begin{aligned}
[[A]_\alpha]^\nu &= [[A]^\nu]_\alpha, \\
[]A[_\alpha]^\nu &=][A]^\nu[_\alpha.
\end{aligned}$$

holds. Namely, an α-cut and a ν-cut are commutative.

Example 2. Suppose $X = \{a, b, c, d\}$ and

$$\begin{aligned}
A &= \{\{0.3, 0.5\}/a, \{0.7\}/b, \{0.9\}/c\}, \\
B &= \{\{0.4, 0.4\}/a, \{0.2, 0.5\}/b\}.
\end{aligned}$$

We can represent A as

$$A = \{(a, 0.3), (a, 0.5), (b, 0.7), (c, 0.9)\}.$$

In the standard form,

$$\begin{aligned} A &= \{(0.5, 0.3)/a, (0.7)/b, (0.9)/c\}, \\ B &= \{(0.4, 0.4)/a, (0.5, 0.2)/b\}. \end{aligned}$$

where zero elements are ignored. We have

$$\begin{aligned} A \cup B &= \{(0.5, 0.4)/a, (0.7, 0.2)/b, (0.9)/c\}, \\ A \cap B &= \{(0.4, 0.3)/a, (0.5)/b\}, \\ A + B &= \{(0.5, 0.4, 0.4, 0.3)/a, (0.7, 0.5, 0.2)/b, (0.9)/c\}, \\ [A]_{0.3} &= \{2/a, 1/b, 1/c\}, \\ ]A[_{0.3} &= \{1/a, 1/b, 1/c\}, \\ [A]^2 &= \{0.3/a\}. \end{aligned}$$

4 Images and α-Cuts

Let f be a mapping of X into Y. Let us consider two images:

$$f(A) = \bigcup_{x \in A} \{f(x)\} \tag{12}$$

and

$$f\langle A \rangle = \sum_{x \in A} \{f(x)\} \tag{13}$$

where A is a fuzzy multiset of X.

Example 3. Suppose

$$A = \{(a, 0.3), (a, 0.5), (b, 0.7), (c, 0.9)\}$$

and $f(a) = v$, $f(b) = f(c) = w$. Then,

$$\begin{aligned} f(A) &= \{(v, 0.5), (w, 0.9)\}, \\ f\langle A \rangle &= \{(v, 0.3), (v, 0.5), (w, 0.7), (w, 0.9)\}. \end{aligned}$$

When A is an ordinary fuzzy set, the former coincides with the ordinary extension principle. Moreover for an arbitrary fuzzy multiset A, $f(A)$ is an ordinary fuzzy set: there is no pair $(x, \nu), (x', \nu') \in f(A)$ such that $x = x'$. Therefore $f(A)$ is inappropriate as a mapping of fuzzy multisets.

For example, let Id be an identity mapping of X onto X: $Id(x) = x$. Suppose A is a fuzzy multiset. Then

$$Id(A) \neq A$$

in general. Instead, we have

$$Id(A) = [A]^1$$

using ν-cut with $\nu = 1$.

On the other hand, the latter can generate a fuzzy multiset from an ordinary fuzzy set. Notice in particular that the latter image uses a simple rewriting of symbols. We thus consider $f\langle A\rangle$ alone hereafter. Remark also that

$$Id\langle A\rangle = A.$$

We have the following.

Proposition 5. For every fuzzy multiset A and B of X,

$$\begin{aligned} f\langle [A]_\alpha\rangle &= [f\langle A\rangle]_\alpha, \\ f\langle]A[_\alpha\rangle &=]f\langle A\rangle[_\alpha, \\ f\langle A+B\rangle &= f\langle A\rangle + f\langle B\rangle. \end{aligned}$$

The proof is omitted.

5 Other Operations

We glimpse other operations for fuzzy multisets of which detailed discussion is omitted.

***t**-Norms*

Let

$$\begin{aligned} C_A(x) &= (\nu_A^1, \dots, \nu_A^p, \dots), \\ C_B(x) &= (\nu_B^1, \dots, \nu_B^p, \dots). \end{aligned}$$

Then,

$$\begin{aligned} C_{A\mathbf{T}B}(x) &= (t(\nu_A^1, \nu_B^1), \dots, t(\nu_A^p, \nu_B^p), \dots), \\ C_{A\mathbf{S}B}(x) &= (s(\nu_A^1, \nu_B^1), \dots, s(\nu_A^p, \nu_B^p), \dots). \end{aligned}$$

In particular,

$$C_{A\cdot B}(x) = (\nu_A^1 \cdot \nu_B^1, \dots, \nu_A^p \cdot \nu_B^p, \dots).$$

Cartesian Product

Let A and B respectively be fuzzy multiset of X and Y:

$$C_A(x) = (\nu_A^1, \nu_A^2, \dots), \qquad C_B(y) = (\nu_B^1, \nu_B^2, \dots),$$

for $x \in X$ and $y \in Y$. Then,

$$C_{A\times B}(x, y) = \{\min\{\nu_A^1, \nu_B^1\}, \min\{\nu_A^1, \nu_B^2\}, \dots, \min\{\nu_A^i, \nu_B^j\}, \dots\}.$$

Thus the multiset for $C_{A\times B}(x, y)$ has every combination of nonzero elements of $C_A(x)$ and $C_B(y)$. For crisp multisets M of X and N of Y, we have

$$\mathcal{I}(M \times N) = \mathcal{I}(M) \times \mathcal{I}(N).$$

Complement of Multiset

We have seen the collection of all fuzzy multisets forms a distributive lattice, the discussion of a complement of a crisp or fuzzy multiset has problems. A way to define a complement is to introduce a value of infinity $+\infty$ and assume $C_M(x) \colon X \to \boldsymbol{N} \cup \{+\infty\}$. It then is not difficult to see that a complement M^C can be defined:

$$C_{M^C}(x) = \begin{cases} 0 & (C_M(x) > 0), \\ +\infty & (C_M(x) = 0) \end{cases}$$

For a fuzzy multiset A, this suggests

$$\begin{aligned} &C_{A^C}(x) = \emptyset \quad (C_A(x) \neq \emptyset), \\ &C_{A^C}(x) = (+\infty, +\infty, \dots) \quad (C_A(x) = \emptyset). \end{aligned}$$

The last definition includes an artificial sequence of $+\infty$ as all elements. Nevertheless, introduction of the infinite element is necessary from the viewpoint of intuitionistic logic, where a complement should be defined to satisfy the axiom of Heyting algebra [21].

For every fuzzy multiset A, the following is valid.

$$\begin{aligned} &A^C \cap A = \emptyset, \\ &A \subseteq (A^C)^C. \end{aligned}$$

6 Spaces of Fuzzy Multisets

We have assumed $C_A(x)$ is finite at the beginning. However, extension to infinite fuzzy multisets is straightforward:

$$C_A(x) = (\nu^1, \dots, \nu^p, \dots)$$

in which we admit infinite nonzero elements. A reasonable assumption to this sequence is

$$\nu^j \to 0 \quad (j \to +\infty).$$

Since this assumption ensures α-cuts $[A]_\alpha$ and $]A[_\alpha$ are finite for all $\alpha \in (0, 1]$.

Metric spaces can be defined on the collections of fuzzy multisets of X. For example, let

$$C_A(x) = (\nu_A^1, \dots, \nu_A^p, \dots), \quad C_B(x) = (\nu_B^1, \dots, \nu_B^p, \dots),$$

Then we can define

$$d_1(A, B) = \sum_{x \in X} \sum_{j=1}^{\infty} |\nu_A^j - \nu_B^j|,$$

which is an ℓ_1 type metric. Moreover we can also define

$$d_2(A, B) = \sqrt{\sum_{x \in X} \sum_{j=1}^{\infty} |\nu_A^j - \nu_B^j|^2},$$

as the ℓ_2 type metric. Moreover a scalar product $< A, B >$ is introduced in the latter space:

$$< A, B > = |A \cdot B| = \sum_{x \in X} |C_{A \cdot B}(x)|$$

using the algebraic product. We have

$$d_2(A, B) = < A, A > + < B, B > -2 < A, B > .$$

It is not difficult to see the metric space with d_1 is a Banach space and that with $< A, B >$ is a Hilbert space [8]. Such metrics are useful in discussing fuzzy multiset model for data clustering [16].

7 Conclusion

Multisets and fuzzy multisets are based on the idea of simple sequential processing of linear lists, as seen in the discussion of $f\langle \cdot \rangle$ which is the sequential rewriting of symbols. Moreover the sorting operation is most important in fuzzy multiset operations.

Spaces in fuzzy multisets are used for clustering and information retrieval [16]. Recent methods in pattern recognition such as the support vector machines [22] can be used in fuzzy multiset space and application to document clustering, which will be shown by us in near future.

Although we have omitted relations between multisets and rough sets [17, 24]. Readers can see, e.g., [15] for the related discussion.

Since a fuzzy multiset represents multiple occurrence of an object with possibly different memberships, it is adequate for information retrieval model especially on the web. There are much room for further studies both in theory and applications.

Acknowledgment

This research has partially been supported by the Grant-in-Aid for Scientific Research (C), Japan Society for the Promotion of Science, No.13680475.

References

1. W. Blizard. Real-valued multisets and fuzzy sets. *Fuzzy Sets and Systems* **33**, 1989, pp. 77–97.
2. C. S. Calude, G. Păun, G. Rozenberg, and A. Salomaa (eds.). *Multiset Processing. Lecture Notes in Computer Science* 2235, Springer, 2001.
3. J. Celko. *Joe Celko's SQL for Smarties: Advanced SQL Programming.* Morgan Kaufmann, 1995.
4. A. di Nola, S. Sessa, W. Pedrycz, and E. Sanchez. *Fuzzy Relation Equations and Their Applications to Knowledge Engineering.* Kluwer, 1989.

5. B. Li, W. Peizhang, and L. Xihui. Fuzzy bags with set-valued statistics. *Computer Math. Applications***15**, 1988, pp. 811–818.
6. D. E. Knuth. *The Art of Computer Programming* vol. 2. Addison-Wesley, 1969.
7. Z. -Q. Liu, S. Miyamoto (eds.). *Soft Computing and Human-Centered Machines.* Springer, Tokyo, 2000.
8. L. A. Lusternik, V. J. Sobolev. *Elements of Functional Analysis.* Hindustan Publishing, Delhi, 1974.
9. Z. Manna, R. Waldinger. *The Logical Basis for Computer Programming* **1**: Deductive Reasoning. Addison-Wesley, 1985.
10. S. Miyamoto. Basic operations of fuzzy multisets. *J. of Japan Society for Fuzzy Theory and Systems* **8: 4**, 1996, pp. 639–645 (in Japanese).
11. S. Miyamoto, K. S. Kim. Multiset-valued images of fuzzy sets, in: *Proceedings of the Third Asian Fuzzy Systems Symposium*, June 18-21, 1998, Masan, Korea, pp.543-548.
12. S. Miyamoto. Application of rough sets to information retrieval. *Journal of the American Society for Information Science* **47: 3**, 1998, pp. 195–205.
13. S. Miyamoto. Rough sets and multisets in a model of information retrieval, in: *Soft Computing in Information Retrieval: Techniques and Applications* (F.Crestani *et al.* (eds.)). Springer, 2000, pp. 373–393.
14. S. Miyamoto. Fuzzy multisets and their generalizations, in:*Multiset Processing* (C.S.Calude *et al.* (eds.)). *Lecture Notes in Computer Science* **2235**. Springer, 2001, pp. 225–235.
15. S. Miyamoto. Generalized multisets and rough approximations,in: *Proceedings of FUZZ-IEEE 2002*, Honolulu, Hawaii, May 12-18, 2002, pp. 751–755.
16. S. Miyamoto. Information clustering based on fuzzy multisets. *Information Processing and Management* **39: 2**, 2003, pp. 195–213.
17. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer, 1991.
18. A. Ramer, C.-C. Wang. Fuzzy multisets, in: *Proceedings of 1996 Asian Fuzzy Systems Symposium*, Dec. 11-14, 1996, Kenting, Taiwan, pp. 429–434.
19. A. Rebai, J.-M. Martel. A fuzzy bag approach to choosing the "best" multiattributed potential actions in a multiple judgement and non cardinal data context. *Fuzzy Sets and Systems* **87**, 1997, pp. 159–166.
20. B. Schweizer, A. Sklar. *Probabilistic Metric Spaces*, North-Holland, 1983.
21. G. Takeuti, S. Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy sets theory. *J. of Symbolic Logic* **3: 3**, 1984, pp. 851–866.
22. V. N. Vapnik. *The Nature of the Statistical Learning Theory* 2nd ed., Springer, 2000.
23. R. R. Yager. On the theory of bags. *Int. J. General Systems* **13**, 1986, pp. 23–37.
24. Y. Y. Yao, S. K. M. Wong, and T. Y. Lin. A review of rough set models, in: *Rough Sets and Data Mining: Analysis of Imprecise Data* (T.Y.Lin, N.Cercone (eds.)), Kluwer, 1997, pp. 47–75.
25. H.-J. Zimmermann. *Fuzzy Set Theory and its Applications* 3rd ed., Kluwer, 1996.

A Non-controversial Definition of Fuzzy Sets

Jaroslav Ramík[1] and Milan Vlach[2]

[1] Institute for Research and Applications of Fuzzy Modelling
University of Ostrava, Czech Republic
[2] Institute of Information Theory and Automation, Praha, Czech Republic

Abstract. Already in the early stages of the development of fuzzy set theory, it has been recognized that fuzzy sets can be defined and represented in several different ways. In this paper we first define fuzzy sets within the classical set theory by suitable nested families of sets, and then we discuss how this definition is related to the usual definition by membership functions.

1 Introduction

A well known fact in the theory of sets is that properties of subsets of a given set X and their mutual relations can be studied by means of their characteristic functions, see, e.g., [2]–[5], [7], and [9]. While this may be advantageous in some contexts, we should notice that the notion of the characteristic function of a subset is more complex than the notion of a subset. Indeed, the characteristic function χ_A of a subset A of X is defined by,

$$\chi_A(x) = \begin{cases} 1 \text{ if } x \in A, \\ 0 \text{ if } x \notin A. \end{cases}$$

Since χ_A is a function, we need not only the underlying set X and its subset A, but also one additional set, in this case the set $\{0, 1\}$ or any other two–element set. Moreover, we also need the notion of the Cartesian product because functions are specially structured binary relations, in this case special subsets of $X \times \{0, 1\}$.

If we define fuzzy sets by means of their membership functions, that is, by replacing the range $\{0, 1\}$ of characteristic functions with a lattice, for example, the unit interval $[0, 1]$ of real numbers with the standard ordering, then we should be aware of the following fact. Such functions may be related to certain objects (build from subsets of the underlying set) in a way analogous to that in which characteristic functions are related to subsets. This may explain why the fuzzy community (rightly?) hesitates to accept the view that a fuzzy subset of a given set is nothing else than its membership function. Then, a natural question arises. Namely, what are those objects? Obviously, it can be expected that they are more complex than just subsets, because the class of functions mapping X into a lattice can be much richer than the class of characteristic functions. In the next section, we show that it is advantageous to define these objects as nested families of subsets satisfying certain mild conditions.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 201–207, 2004.

It should be noted that fuzzy sets and membership functions are closely related to the inherent imprecision of linguistic expressions in natural languages. Applying classical set theory it requires an artificially derived threshold to determine, whether, e.g., a given person belongs to the set of "tall persons" or not. Probability theory does not provide a way out, as it usually deals with crisp events and the uncertainty is about whether this event will occur or not. However, in fuzzy logic, this is the question of degree of truth rather than a simple "yes or no" decision.

2 Definition and Basic Properties

In order to define the concept of a fuzzy subset of a given set X within the framework of standard set theory we implicitly employ the concept of upper level set of a function. Throughout this paper, X is a nonempty set.

Definition 1. *Let X be a set, a* fuzzy subset A of X *is the family of subsets $A_\alpha \subset X$, where $\alpha \in [0,1]$, satisfying the following properties:*

$$\begin{array}{ll} \textit{(i)} & A_0 = X, \\ \textit{(ii)} & A_\beta \subset A_\alpha \ \textit{whenever}\ 0 \le \alpha \le \beta \le 1, \\ \textit{(iii)} & A_\beta = \bigcap\limits_{0\le\alpha<\beta} A_\alpha. \end{array} \tag{1}$$

A fuzzy subset A of X will be also called a fuzzy set.

The membership function $\mu_A : X \to [0,1]$ of a fuzzy set A *is defined as follows,*

$$\mu_A(x) = \sup\{\alpha \mid \alpha \in [0,1], x \in A_\alpha\}, \tag{2}$$

for each $x \in X$.

The core of A, *$Core(A)$, is given by,*

$$Core(A) = \{x \in X \mid \mu_A(x) = 1\}. \tag{3}$$

A fuzzy subset A of X is called normalized *when its core is nonempty.*

The support of A, *$Supp(A)$, is given by,*

$$Supp(A) = Cl(\{x \in X \mid \mu_A(x) > 0\}). \tag{4}$$

The height of A, *$Hgt(A)$, is given by,*

$$Hgt(A) = \sup\{\mu_A(x) \mid x \in X\}. \tag{5}$$

For each $\alpha \in [0,1]$, the α-cut of A, *$[A]_\alpha$, is given by*

$$[A]_\alpha = \{x \in X \mid \mu_A(x) \ge \alpha\}. \tag{6}$$

For $x \in X$, the function value $\mu_A(x)$ is called the membership degree of x in the fuzzy set A. *The class of all fuzzy subsets of X is denoted by $\mathcal{F}(X)$.*

Let $\mu : X \to [0, 1]$ be an arbitrary function, $\alpha \in [0, 1]$. The upper level set $U(\mu, \alpha)$ of μ at α is defined as follows

$$U(\mu, \alpha) = \{x \in X \mid \mu(x) \geq \alpha\}. \tag{7}$$

In the following two propositions, we show that the family generated by the upper level sets of μ satisfies conditions (1), thus, it generates a fuzzy subset of X and the membership function μ_A defined by (2) coincides with μ. Moreover, for a given fuzzy set $A = \{A_\alpha\}_{\alpha\in[0,1]}$, every α–cut $[A]_\alpha$, given by (6) coincides with the corresponding A_α.

Proposition 1. *Let $\mu : X \to [0, 1]$ be a function and let $A = \{A_\alpha\}_{\alpha\in[0,1]}$ be a family of its upper–level sets, i.e., $A_\alpha = U(\mu, \alpha)$ for all $\alpha \in [0, 1]$. Then A is a fuzzy subset of X and μ is a membership function of A.*

Proof. First, we prove that $A = \{A_\alpha\}_{\alpha\in[0,1]}$ satisfies conditions (1), where $A_\alpha = U(\mu, \alpha)$ for all $\alpha \in [0, 1]$. Indeed, conditions (i) and (ii) hold easily. For condition (iii), we observe that by (ii) it follows that $A_\beta \subset \bigcap_{0\leq\alpha<\beta} A_\alpha$. To prove the opposite inclusion, let

$$x \in \bigcap_{0\leq\alpha<\beta} A_\alpha. \tag{8}$$

Assume the contrary, let $x \notin A_\beta$. Then $\mu(x) < \beta$ and there exists α', such that $\mu(x) < \alpha' < \beta$. Then $x \in A_{\alpha'}$. This contradicts (8). Hence $x \in A_\beta$.

It remains to prove that $\mu = \mu_A$, where μ_A is the membership function of A. For this purpose let $x \in X$ and let us show that $\mu(x) = \mu_A(x)$.

By definition (2) we have

$$\begin{aligned} &\mu_A(x) = \sup\{\alpha \mid \alpha \in [0, 1], x \in A_\alpha\} = \sup\{\alpha \mid \alpha \in [0, 1], x \in U(\mu, \alpha)\} \\ &= \sup\{\alpha \mid \alpha \in [0, 1], \mu(x) \geq \alpha\}, \end{aligned} \tag{9}$$

therefore, $\mu(x) = \mu_A(x)$.

Proposition 2. *Let $A = \{A_\alpha\}_{\alpha\in[0,1]}$ be a fuzzy subset of X and let $\mu_A : X \to [0, 1]$ be the membership function of A. Then for all $\alpha \in [0, 1]$, the α–cuts $[A]_\alpha$ coincide with A_α, i.e. $[A]_\alpha = A_\alpha$.*

Proof. Let $\beta \in [0, 1]$. By (2), observe that $A_\beta \subset [A]_\beta$. It suffices to prove the opposite inclusion.

Let $x \in [A]_\beta$. Then $\mu_A(x) \geq \beta$, or, equivalently, $\sup\{\alpha \mid \alpha \in [0, 1], x \in A_\alpha\} \geq \beta$. By monotonicity condition (ii) in (1) it follows that we have $x \in A_\alpha$ for all $0 \leq \alpha \leq \beta$. Hence, $x \in \bigcap_{0\leq\alpha<\beta} A_\alpha$, however, applying (iii) in (1) we get $x \in A_\beta$. Consequently, $[A]_\beta \subset A_\beta$.

The above stated results allow for introducing a natural one–to–one correspondence between fuzzy subsets of X and real functions mapping X to $[0, 1]$. Each fuzzy subset A of X is given by its membership function μ_A and vice–versa, each function $\mu : X \to [0,1]$ uniquely determines a fuzzy subset A of X, with the property that the membership function μ_A of A is μ.

The notions of inclusion and equality can be extended to fuzzy subsets as follows. Let A and B be fuzzy subsets of X, $A = \{A_\alpha\}_{\alpha\in[0,1]}$, $B = \{B_\alpha\}_{\alpha\in[0,1]}$.

$$A \subset B \text{ if } A_\alpha \subset B_\alpha \text{ for each } \alpha \in [0,1], \tag{10}$$

$$A = B \text{ if } A_\alpha = B_\alpha \text{ for each } \alpha \in [0,1]. \tag{11}$$

The following property is obvious.

Proposition 3. *Let A and B be fuzzy subsets of X, $A = \{A_\alpha\}_{\alpha\in[0,1]}$, $B = \{B_\alpha\}_{\alpha\in[0,1]}$. Then the following holds,*

$$A \subset B \textit{ if and only if } \mu_A(x) \le \mu_B(x) \textit{ for all } x \in X, \tag{12}$$

$$A = B \textit{ if and only if } \mu_A(x) = \mu_B(x) \textit{ for all } x \in X, \tag{13}$$

$$A = B \textit{ if and only if } A \subset B \textit{ and } B \subset A.$$

Proof. We prove only (12), the proof of statement (13) is analogous. Let $x \in X$, $A \subset B$. Then by definition (10), $A_\alpha \subset B_\alpha$ for each $\alpha \in [0,1]$. Using (2), we obtain,

$$\begin{aligned}\mu_A(x) &= \sup\{\alpha \mid \alpha \in [0,1], x \in A_\alpha\}\\ &\le \sup\{\alpha \mid \alpha \in [0,1], x \in B_\alpha\} = \mu_B(x).\end{aligned}$$

Suppose that $\mu_A(x) \le \mu_B(x)$ holds for all $x \in X$ and let $\alpha \in [0,1]$. We have to show that $A_\alpha \subset B_\alpha$. Indeed, for an arbitrary $u \in A_\alpha$,

$$\sup\{\beta \mid \beta \in [0,1], u \in A_\beta\} \le \sup\{\beta \mid \beta \in [0,1], u \in B_\beta\}.$$

From this, $\sup\{\beta \mid \beta \in [0,1], u \in B_\beta\} \ge \alpha$, therefore, from the monotonicity (ii) of (1) for each $\beta < \alpha$ it follows that $u \in B_\beta$. Hence, by (iii) in Definition 1, we obtain $u \in B_\alpha$.

The last part of the proof is obvious.

The classical subsets of a given set can be considered as special fuzzy subsets whose defining families consists of the same elements. We obtain the following definition.

Definition 2. *A fuzzy subset $\{A_\alpha\}_{\alpha\in[0,1]}$ of X is said to be a* crisp subset of X*, if there exists $\mathsf{A} \subset X$ such that $A_\alpha = \mathsf{A}$ for every $\alpha \in (0,1]$.*

Obviously, such fuzzy set is uniquely determined by A, in this case we write $A = \{\mathsf{A}\}_{\alpha\in[0,1]}$. From this definition and the definition of the equality of fuzzy subsets, it is obvious that there is a natural one–to–one correspondence between crisp subsets of X and (classical) subsets of X. This makes it possible to embed the set $\mathcal{P}(X)$ of all subsets of X into the set $\mathcal{F}(X)$ of all fuzzy subsets of X. The following proposition clarifies how the membership functions of crisp subsets of X are related to the characteristic functions of subsets of X.

Proposition 4. *Let* A *be a nonempty subset of* X *and let* A *be the crisp subset of* X *corresponding to* A *in the natural embedding of* $\mathcal{P}(X)$ *into* $\mathcal{F}(X)$*. Then the membership function of* A *is equal to to the characteristic function of* A*.*

Proof. Let $x \in X, x \notin \mathsf{A}$. Then by definition (2) and Def. 2, $\mu_A(x) = \sup\{\alpha \mid \alpha \in [0,1], x \in A_\alpha\} = 0 = \chi_{\mathsf{A}}(x)$.

Let $x \in X, x \in \mathsf{A}$. Then again by definition (2) and Def. 2, $\mu_A(x) = \sup\{\alpha \mid \alpha \in [0,1], x \in A_\alpha\} = \sup\{\alpha \mid \alpha \in [0,1], x \in \mathsf{A}\} = 1 = \chi_{\mathsf{A}}(x)$.

Example 1. Let $\mu : \mathbf{R} \to [0,1]$ be defined by $\mu(x) = e^{-x^2}$ for $x \in \mathbf{R}$. Let $A' = \{A'_\alpha\}_{\alpha\in[0,1]}$, $A'' = \{A''_\alpha\}_{\alpha\in[0,1]}$ be two families of subsets in $\mathbf{R}$ defined as follows,

$$A'_\alpha = \begin{cases} \mathbf{R} & \text{if } \alpha = 0, \\ \{x \mid x \in \mathbf{R}, \mu(x) > \alpha\} & \text{if } 0 < \alpha \le 1, \end{cases}$$

$$A''_\alpha = \begin{cases} \mathbf{R} & \text{if } \alpha = 0, \\ \{x \mid x \in \mathbf{R}, \mu(x) \ge \alpha\} & \text{if } 0 < \alpha \le 1. \end{cases}$$

Clearly $A' \neq A''$. Observe that (i) and (ii) are satisfied for A' and A'', however $A'_1 = \emptyset$, $\bigcap\limits_{0\le\alpha<1} A'_\alpha = \{0\}$, thus (iii) in (1) is not satisfied. Hence A' is not a fuzzy subset of $\mathbf{R}$. On the other hand, (iii) in (1) is satisfied with A''. Hence A'' is a fuzzy set.

3 Operations with Fuzzy Sets

In order to generalize the Boolean set–theoretical operations for fuzzy sets: intersection, union and complement, it is natural to use triangular norms, triangular co–norms and fuzzy negations, see [7].

Let T be a t–norm, S be a t–conorm, and N be a fuzzy negation. Furthermore, let μ_A and μ_B be the membership functions of fuzzy subsets A and B of X, respectively. We introduce the operations of *intersection* $\cap_T$, *union* $\cup_S$ and *complement* $\mathcal{C}_N$, on $\mathcal{F}(X)$, by defining the membership functions of the fuzzy subsets $A \cap_T B$, $A \cup_S B$ and $\mathcal{C}_N A$ of X as follows (see [9]),

$$\begin{aligned} \mu_{A\cap_T B}(x) &= T(\mu_A(x), \mu_B(x)), \\ \mu_{A\cup_S B}(x) &= S(\mu_A(x), \mu_B(x)), \\ \mu_{\mathcal{C}_N A}(x) &= N(\mu_A(x)). \end{aligned} \tag{14}$$

The operations introduced by L. Zadeh in [1] have been originally based on $T = T_M = \min$, $S = S_M = \max$, and standard negation N defined for each $\alpha \in [0,1]$ by $N(\alpha) = 1 - \alpha$. The properties of the operations intersection $\cap_T$, union $\cup_S$ and complement $\mathcal{C}_N$ can be derived directly from the corresponding properties of t–norm T, t–conorm S and fuzzy negation N.

Notice that for all $A \in \mathcal{F}(X)$, we do not necessarily obtain properties which hold for crisp sets, namely,

$$A \cap_T \mathcal{C}_N A = \emptyset, \tag{15}$$

$$A \cup_S \mathcal{C}_N A = X. \tag{16}$$

If t–norm T in the De Morgan triple (T, S, N) does not have zero divisors, e.g., $T = \min$, then these properties never hold unless A is a crisp set. On the other hand, for the De Morgan triple (T_L, S_L, N) based on Łukasiewicz t–norm $T = T_L$, properties (15) are satisfied, see [7].

Given a t–norm T and fuzzy subsets A and B of X and Y, respectively, the *Cartesian product* $A \times_T B$ is the fuzzy subset of $X \times Y$ with the following membership function,

$$\begin{aligned} &\mu_{A\times_T B}(x, y) = T(\mu_A(x), \mu_B(y)) \\ &\qquad \text{for } (x, y) \in X \times Y. \end{aligned} \tag{17}$$

An interesting and natural question arises, whether the α–cuts of the intersection $A \cap_T B$, the union $A \cup_S B$ and the Cartesian product $A \times_T B$ of $A, B \in \mathcal{F}(X)$, coincide with the intersection, union and the Cartesian product, respectively, of the corresponding α-cuts $[A]_\alpha$ and $[B]_\alpha$. We have the following result, see [7].

Proposition 5. *Let T be a t–norm, S be a t–conorm, $\alpha \in [0, 1]$. Then, the equalities,*

$$\begin{aligned} [A \cap_T B]_\alpha &= [A]_\alpha \cap [B]_\alpha, \\ [A \cup_S B]_\alpha &= [A]_\alpha \cup [B]_\alpha, \\ [A \times_T B]_\alpha &= [A]_\alpha \times [B]_\alpha, \end{aligned} \tag{18}$$

hold for all fuzzy sets $A, B \in \mathcal{F}(X)$ if and only if α is an idempotent element of both T and S.

In particular, this result means that identities (18) hold for all $\alpha \in [0, 1]$ and for all fuzzy sets $A, B \in \mathcal{F}(X)$ if and only if $T = T_M$ and $S = S_M$. For the proof of Proposition 5 and more details we refere to [7].

4 Conclusion

In this paper the definition of fuzzy sets by a family of level sets has been presented. Its equivalence and advantages in comparison with the standard definition by the membership function have been discussed. We have shown that by the definition of fuzzy sets based on the family of sets, a consistent theory of fuzzy sets equivalent to the standard one can be built up. Moreover, crisp sets can be naturally embedded into the fuzzy sets and no “artificial” identification of a fuzzy set with its membership function is necessary. We hope that this approach may help to overcome a misunderstanding between the fuzzy community that is reluctant to accept the view that a fuzzy subset of a given set is nothing else than its membership function on one side and the community of mathematicians - functional analysts on the other side.

References

1. L.A. Zadeh. Fuzzy sets. *Information and Control* **8**, 1965, pp. 338–353.
2. R. Bellmann, L.A. Zadeh. Decision making in fuzzy environment. *Management Science* **17: 4**, 1970, pp. 141–164.
3. D. Dubois, H. Prade. *Possibility theory.* Plenum Press, 1988.
4. D. Dubois et al. Fuzzy interval analysis, in: *Fundamentals of Fuzzy Sets.* Kluwer Series on Fuzzy Sets, vol. 1, Kluwer, 2000.
5. J.C. Fodor, M. Roubens. *Fuzzy Preference Modelling and Multi–Criteria Decision Support.* Kluwer, 1994.
6. P. Hájek. *Metamathematics of Fuzzy Logic.* Kluwer Series Trends in Logic, Kluwer, 1998.
7. E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms.* Kluwer, 2000.
8. D. Ralescu. A survey of the representation of fuzzy concepts and its applications, in: *Advances in Fuzzy Sets Theory and Applications* (M.M. Gupta, R.K. Regade, and R. Yager (eds.)). North Holland, 1979, pp. 77–91.
9. J. Ramik, M. Vlach. *Generalized Concavity in Fuzzy Optimization and Decision Analysis.* Kluwer, 2001.

Algebraic Structures for Rough Sets

Gianpiero Cattaneo and Davide Ciucci

Dipartimento Di Informatica, Sistemistica e Comunicazione
Università di Milano, Bicocca
Via Bicocca degli Arcimboldi 8, I-20126 Milano, Italy
{cattang,ciucci}@disco.unimib.it

Abstract. Using as example an incomplete information system with support a set of objects X, we discuss a possible algebraization of the concrete algebra of the power set of X through quasi BZ lattices. This structure enables us to define two rough approximations based on a similarity and on a preclusive relation, with the second one always better that the former. Then, we turn our attention to Pawlak rough sets and consider some of their possible algebraic structures. Finally, we will see that also Fuzzy Sets are a model of the same algebras. Particular attention is given to HW algebra which is a strong and rich structure able to characterize both rough sets and fuzzy sets.

1 Introduction

Rough structures describe the behavior of *concepts*, *properties*, *data*, *abstract objects* in general, that may present some intrinsic vague, ambiguous, non–sharp features. For the sake of simplicity we will simply speak of *vague objects*. To these objects it is usually associated some data which characterize them and which can be used to classify them. Data that are often organized as a table associating to each object some attribute values. As observed by Pawlak [Paw98]:

> Data are often presented as a table, columns of which are labeled by *attributes*, rows by *objects* of interest and entries of the table are *attribute values*.
> For example, in a table containing information about patients suffering from a certain disease objects are *patients* (strictly speaking their ID's), attributes can be for example *blood pressure*, *body temperature* etc., whereas the entry corresponding to object Smiths and the attribute *blood pressure* can be *normal*.
> Such tables are known as *information systems*, *attribute-value tables* or *information tables*.

A natural abstract definition, that reflects this behavior, is the following one.

Definition 1. *An* Information System *is a structure*

$$\mathcal{K}(X) = \langle X,\, Att(X),\, val(X),\, F\rangle$$

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 208–252, 2004.

where:

- *X (called the* universe*) is a non empty set of* objects *(*situations*,* entities*,* states*);*
- *$Att(X)$ is a non empty set of* attributes*, which assume values for the objects belonging to the set X;*
- *$val(X)$ is the set of all* possible values *that can be observed for an attribute a from $Att(X)$ in the case of an object x from X;*
- *F (called the* information map*) is a mapping $F : X \times Att(X) \to val(X)$ which associates to any pair, consisting of an object $x \in X$ and of an attribute $a \in Att(X)$, the value $F(x, a) \in val(X)$ assumed by a for the object x.*

In general we assume that an information system satisfies the following two conditions of coherence:

1. *F must be surjective; this means that if there exists a value $v \in val(X)$ which is not the result of the application of the information map F to any pair $(x, a) \in X \times Att(X)$, then this value has no interest with respect to the knowledge stored in the information system.*
2. *For any attribute $a \in Att(X)$ there exist at least two objects x_1 and x_2 such that $F(x_1, a) \neq F(x_2, a)$, otherwise this attribute does not supply any knowledge and can be suppressed.*

Example 1. Imagine you want to rent a flat, and you start to collect information about some apartments. The features you are interested in are: the price of the flat; its location, i.e., if it is down–town or not; the number of rooms and if it has furniture or not. But you are not interested in, for example, at which floor it is located. So, when organizing the data in your possession, you will consider just the first four attributes and omit the floor number attribute. The result is a situation similar to the one presented in Table 1, where the set of objects is $X = \{f_1, f_2, f_3, f_4, f_5, f_6\}$, the family of attributes is $Att(X)$={Price, Rooms, Down-Town, Furniture} and the set of all possible values is $val(X)$={high, low, medium, 1, 2, yes, no}.

Table 1. Flats information system.

Flat	Price	Rooms	Down–Town	Furniture
f_1	high	2	yes	no
f_2	high	1	yes	no
f_3	high	2	yes	no
f_4	low	1	no	no
f_5	low	1	no	no
f_6	medium	1	yes	yes

A generalization of such a concept are *incomplete information systems*, i.e., information systems in which not all values are available for all objects. Such a generalization is justified by empirical observations. Quoting [ST99]:

> An explicit hypothesis done in the classic rough set theory is that all available objects are completely described by the set of available attributes. [...]
> Such a hypothesis, although sound, contrasts with several empirical situations where the information concerning the set A is only partial either because it has not been possible to obtain the attribute values or because it is definitely impossible to get a value for some object on a given attribute.

So, let $\mathcal{K}(X) = \langle X, Att(X), Val(X), F\rangle$ be an incomplete information system. In order to denote the fact that the value possessed by an object x_i with respect to the attribute a_j is unknown, we introduce a *null value* and write $F(x_i, a_j) = *$.

Example 2. As a concrete example, let us consider the information system described in Table 1. It can happen that given a flat we do not know all its features, for instance because some information was missing on the advertisement. The result is some missing values in the information system, as shown in Table 2.

Table 2. Flats incomplete information system.

Flat	Price	Rooms	Down–Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	yes	no
f_4	low	*	no	no
f_5	low	1	*	no
f_6	*	1	yes	*

Given an information system, complete or not, if we consider pairs of objects belonging to the universe X, we can describe their relationship through a binary relation $\mathcal{R}$. A classification and logical–algebraic characterization of such binary relations can be found in literature (for an overview see [Orl98b]). Generally, these relations are divided into two groups: *indistinguishability* and *distinguishability* relations. In our analysis, we are dealing with a *tolerance* (or *similarity*) relation, i.e., a reflexive and symmetric relation, and its opposite, a *preclusivity* relation, i.e., an irreflexive and symmetric relation. From the intuitive point of view, two individuals are similar when they have an "indistinguishable role" with respect to the intended application, even if they are not equivalent. On the other side, they are preclusive if they have a "distinguishable role", even if they are not totally different.

These considerations lead to the definition of similarity space.

Definition 2. *A* similarity space *is a structure $\mathcal{S} = \langle X, \mathcal{R}\rangle$, where X (called the* universe *of the space) is a non–empty set of objects and $\mathcal{R}$ (called the* similarity *relation of the space) is a reflexive and symmetric binary relation defined on X. In other words:*

(i) $\forall x \in X : x\mathcal{R}x$ *(reflexivity);*
(ii) $\forall x, y \in X : x\mathcal{R}y$ *implies* $y\mathcal{R}x$ *(symmetry).*

Let us stress that, in general, a similarity relation is not required to be transitive. This feature strongly differentiates its behavior with respect to the equivalence relation, which is at the basis of classical rough sets theory.

In the context of an incomplete information system $\mathcal{K}(X)$, for a fixed set of attributes $D \subseteq Att(X)$ a natural similarity relation is that two objects are similar if they possess the same values with respect to all known attributes inside D. In a more formal way:

$$\begin{aligned} \forall x, y \in X : \quad & x\mathcal{R}_D y \quad \text{iff} \quad \forall a_i \in D \subseteq Att(X), \\ & \text{either} \quad F(x, a_i) = F(y, a_i) \quad \text{or} \quad F(x, a_i) = * \quad \text{or} \quad F(y, a_i) = * \end{aligned} \tag{1}$$

This is the approach introduced by Kryszkiewicz in [Kry98] which has the advantage that the possibility of null values "corresponds to the idea that such values are just missing, but they do exist. In other words, it is our imperfect knowledge that obliges us to work with a partial information table" [ST99].

Example 3. Considering the information system in Table 2, we have, for example, that f_4 is similar to f_5 whatever be the chosen subset $\mathcal{D}$ of attributes. The same applies to f_5 and f_6. On the other hand, the pair of objects f_4 and f_6 are not similar relatively to any subset of attributes which contains the attribute Down–Town; indeed $F(Down - -Town, f_4) = no$ whereas $F(Down - -Town, f_6) = yes$. In this way, it is also verified that the similarity relation induced from any set of attributes $\mathcal{D}$ that contains Down–Town is not transitive since f_4 is similar to f_5 and f_5 is similar to f_6, but f_4 is not similar to f_6 relatively to $\mathcal{D}$.

Given a similarity space $\langle X, \mathcal{R} \rangle$, the *similarity class* generated by the element $x \in X$ is the collection of all objects similar to x, i.e.,

$$S(x) := \{y \in X : x\mathcal{R}y\}$$

Thus, the similarity class generated by x consists of all the elements which are indistinguishable from x with respect to the similarity relation $\mathcal{R}$. In this way this class constitute a *granule of similarity knowledge* about x and is also called the *granule* generated by x.

Trivially, $\mathcal{R}$ being a reflexive relation, it holds that $x \in S(x)$ and so no similarity class is empty. Further, as a consequence of the non–transitivity of the relation, the similarity classes are not necessarily disjoint, i.e., there may exist x, y such that $S(x) \neq S(y)$ and $S(x) \cap S(y) \neq \emptyset$.

Example 4. Making reference to example 2, if one considers the set of all attributes (i.e., $D = Att(X)$) and the induced similarity relation according to (1) we have that $S(f_4) = \{f_4, f_5\}$ and $S(f_6) = \{f_2, f_5, f_6\}$. Thus, $S(f_4) \neq S(f_6)$ and $S(f_4) \cap S(f_6) = \{f_5\}$.

Using this notion of similarity class, it is possible to define in a natural way a rough approximation by similarity of any set of objects ([Vak91,SS96], [Ste98,ST01]).

Definition 3. *Given a similarity space $\langle X, \mathcal{R}\rangle$, and a set of objects $A \subseteq X$, the* rough approximation of A by similarity *is defined as the pair $\langle \mathbf{L}_{\mathcal{R}}(A), \mathbf{U}_{\mathcal{R}}(A)\rangle$, where*

$$\mathbf{L}_{\mathcal{R}}(A) := \{x \in X : S(x) \subseteq A\} = \{x \in X : \forall z\ (x\mathcal{R}z \Rightarrow z \in A)\} \tag{2a}$$
$$\mathbf{U}_{\mathcal{R}}(A) := \{x \in X : S(x) \cap A \neq \emptyset\} = \{x \in X : \exists z\ (x\mathcal{R}z \textit{ and } z \in A)\} \tag{2b}$$

It is easy to verify that the following chain of inclusions holds:

$$\mathbf{L}_{\mathcal{R}}(A) \subseteq A \subseteq \mathbf{U}_{\mathcal{R}}(A). \tag{3}$$

$\mathbf{L}_{\mathcal{R}}(A)$ *is called the* similarity lower *approximation of A (since it is constituted by all objects whose granule is contained in A) and $\mathbf{U}_{\mathcal{R}}(A)$ the* similarity upper *approximation of A (since it is constituted by all objects whose granule has at least a point in common with A).*

Example 5. Consider again the information system in Table 2 with similarity relation (1) relative to $D = Att(X)$. Choosing the set of objects $A = \{f_4, f_5, f_6\}$, we have $\mathbf{L}_{\mathcal{R}}(A) = \{f_4, f_5\}$ and $\mathbf{U}_{\mathcal{R}}(A) = \{f_2, f_4, f_5, f_6\}$ with the obvious satisfaction of the chain (3).

As said before, the opposite of a similarity relation is a preclusive relation: two objects are in a preclusive relation iff it is possible to distinguish one from the other. Using such a relation it is possible to define a notion dual to the one of similarity space.

Definition 4. *A* preclusivity space *is a structure $\mathcal{S} = \langle X, \#\rangle$, where X (called the* universe *of the space) is a non empty set and $\#$ (called the* preclusivity *relation of the space) is an irreflexive and symmetric relation defined on X. In other words:*

(i) $\forall x \in X :$ *not* $x\#x$ *(irreflexivity);*
(ii) $\forall x, y \in X : x\#y$ *implies* $y\#x$ *(symmetry).*

Needless to stress, any similarity space $\langle X, \mathcal{R}\rangle$ determines a corresponding preclusivity space $\langle X, \#_{\mathcal{R}}\rangle$ with $x\#_{\mathcal{R}}y$ iff $\neg(x\mathcal{R}y)$, and vice versa any preclusivity space $\langle X, \#\rangle$ determines a similarity space $\langle X, \mathcal{R}_{\#}\rangle$ with $x\mathcal{R}_{\#}y$ iff $\neg(x\#y)$. In this case we will say that we have a pair of *correlated* similarity–preclusive relations.

Suppose, now, a preclusivity space $\langle X, \#\rangle$. The preclusive relation $\#$ permits us to introduce for all $H \in \mathcal{P}(X)$ (where we denote by $\mathcal{P}(X)$ the power set of X) its *preclusive complement* defined as

$$H^{\#} := \{x \in X : \forall y \in H\ (x\#y)\}.$$

In other words, $H^{\#}$ contains all and only the elements of X that are distinguishable from all the elements of H. Whenever $x \in H^{\#}$ we will also write: $x\#H$ and we will say that x is *preclusive* to the set H. Further, we will say that two subsets H and K of X are *mutually preclusive* ($H\#K$) iff all the elements of H are preclusive to K, and all the elements of K are preclusive to H. We remark that, in the context of modal analysis of rough approximation spaces, the operation $^{\#}$ is a *sufficiency* operator [DO01].

The following result has been proved in [Cat97].

Proposition 1. *Let $\mathcal{S} = \langle X, \# \rangle$ be a preclusivity space and let us consider the structure $\langle \mathcal{P}(X), \cap, \cup, {}^c, {}^{\#}, \emptyset, X \rangle$. Then, the following hold:*

1. *The substructure $\langle \mathcal{P}(X), \cap, \cup, \emptyset, X \rangle$ is a distributive (complete) lattice, with respect to standard set theoretic intersection $\cap$ and union $\cup$, bounded by the least element $\emptyset$ and the greatest element X. The partial order relation induced from this lattice structure is the standard set theoretic inclusion $\subseteq$.*
2. *The operation ${}^c : \mathcal{P}(X) \to \mathcal{P}(X)$ associating to any subset H of X its set theoretic complement $H^c = X \setminus H$ is a standard complementation, i.e., it satisfies:*

   (C-1) $H = (H^c)^c$ *(involution)*

   (C-2a) $H \subseteq K$ iff $K^c \subseteq H^c$ *(contraposition)*
   (C-2b) $H^c \cap K^c = (H \cup K)^c$ *($\cap$ de Morgan)*
   (C-2c) $H^c \cup K^c = (H \cap K)^c$ *($\cup$ de Morgan)*

   (C-3a) $H \cap H^c = \emptyset$ *(non contradiction)*
   (C-3b) $H \cup H^c = X$ *(excluded middle)*
3. *The operator ${}^{\#} : \mathcal{P}(X) \to \mathcal{P}(X)$ associating to any subset H of X its preclusive complement $H^{\#}$ is a unusual complementation and it satisfies:*
   (B1) $H \subseteq (H^{\#})^{\#}$ *(weak double negation)*

   (B2a) If $H \subseteq K$ *then* $K^{\#} \subseteq H^{\#}$ *(contraposition)*
   (B2b) $H^{\#} \cap K^{\#} = (H \cup K)^{\#}$ *($\cap$ de Morgan for $\#$)*

   (B3) $H \cap H^{\#} = \emptyset$ *(non contradiction)*
4. *The following interconnection rule holds:*
   (in) $H^{\#} \subseteq H^c$.

In the particular case of the similarity relation (1), the corresponding preclusive relation is just its logical negation:

$$\forall x, y \in X \quad x \#_D y \quad \text{iff} \quad \text{not } x \mathcal{R}_D y \quad \text{iff}$$
$$\exists a_i \in D \subseteq Att(X) : F(x, a_i) \neq F(y, a_i) \quad \text{and}$$
$$F(x, a_i) \neq * \quad \text{and} \quad F(y, a_i) \neq * \qquad (4)$$

Example 6. Consider, now, the set of flats $H = \{f_2, f_6\}$ relatively to Table 2, then the preclusive complement of H with respect to $D = Att(X)$ is $H^{\#} = \{f_4\}$, which is different from $H^c = \{f_1, f_3, f_4, f_5\}$. Further, the complementation $^{\#}$ does not satisfy some standard properties. First of all the strong double negation law, $H = H^{\#\#}$, and the excluded middle law $H \cup H^{\#} = X$. In fact, we have

$$H = \{f_2, f_6\} \subsetneqq \{f_1, f_2, f_3, f_6\} = H^{\#\#}$$

and

$$H \cup H^{\#} = \{f_2, f_4, f_6\} \neq X$$

Moreover, also the $\cup$ de Morgan law, $H^{\#} \cup K^{\#} = (H \cap K)^{\#}$, and the strong contraposition law, $H^{\#} \subseteq K^{\#}$ implies $K \subseteq H$, are not generally verified. Let us consider, for instance, $K = \{f_1, f_3\}$. Then, we obtain

$$(H \cap K)^{\#} = \emptyset^{\#} = X \neq \{f_4, f_5, f_6\} = H^{\#} \cup K^{\#}$$

and

$$H^{\#} = \{f_4\} \subseteq \{f_4, f_5, f_6\} = K^{\#} \quad \text{but} \quad K = \{f_1, f_3\} \not\subseteq \{f_2, f_6\} = H$$

Finally, in general, it does not hold the strong interconnection rule $H^{\#c} = H^{\#\#}$. For instance, $H^{\#\#} = \{f_1, f_2, f_3, f_6\} \neq \{f_1, f_2, f_3, f_5, f_6\} = H^{\#c}$.

On the Boolean lattice based on the power set $\mathcal{P}(X)$, we now have two, generally different, complementations: the usual set theoretic complementation ^{c} and the preclusive complementation ${}^{\#}$. By their interaction, it is possible to define a closure and an interior operator on $\mathcal{P}(X)$.

Proposition 2. *Let $\langle \mathcal{P}(X), \cap, \cup, {}^c, {}^{\#}, \emptyset, X \rangle$ be the algebraic structure based on the power set of X and generated by the preclusivity space $\langle X, \# \rangle$. Then the mapping*

$$\mathbf{L}_{\#} : \mathcal{P}(X) \to \mathcal{P}(X), \quad H \to \mathbf{L}_{\#}(H) := H^{c\#\#c}$$

is an interior operator*, i.e.,*

$$\begin{array}{lll}
(I_0) & X = \mathbf{L}_{\#}(X) & (\mathit{normalized}) \\
(I_1) & \mathbf{L}_{\#}(H) \subseteq H & (\mathit{decreasing}) \\
(I_2) & \mathbf{L}_{\#}(H) = \mathbf{L}_{\#}(\mathbf{L}_{\#}(H)) & (\mathit{idempotent}) \\
(I_3) & \mathbf{L}_{\#}(H \cap K) \subseteq \mathbf{L}_{\#}(H) \cap \mathbf{L}_{\#}(K) & (\mathit{sub-multiplicative})
\end{array}$$

Under condition $(I_0) - (I_1)$, the condition (I_3) is equivalent to the *monotonicity condition*: $H \subseteq K$ implies $\mathbf{L}_{\#}(H) \subseteq \mathbf{L}_{\#}(K)$. In general, the Kuratowsky property of topological interior, $\mathbf{L}_{\#}(H \cap K) = \mathbf{L}_{\#}(H) \cap \mathbf{L}_{\#}(K)$, is not verified.

Example 7. Let us consider the two sets of flats $H = \{f_1, f_2, f_4, f_5\}$ and $K = \{f_1, f_2, f_5, f_6\}$, relative to Table 2. Then, the interior of H, K and their intersection, relative to $D = \mathit{Att}(X)$, are respectively $\mathbf{L}_{\#}(H) = \{f_4, f_5\}$, $\mathbf{L}_{\#}(K) = \{f_2, f_5, f_6\}$ and $\mathbf{L}_{\#}(\{f_1, f_2, f_5\}) = \emptyset$. So, $\mathbf{L}_{\#}(H \cap K) = \emptyset \neq \{f_5\} = \mathbf{L}_{\#}(H) \cap \mathbf{L}_{\#}(K)$.

From the fact that according to (I_1) one has that $\mathbf{L}_{\#}(H) \subseteq H$, it is possible to single out the collection of all $\#$–*open* sets defined as follows:

$$\mathbb{O}(X, \#) := \{A \subseteq X : A = \mathbf{L}_{\#}(A) = A^{c\#\#c}\}.$$

This set is not empty since both the empty set $\emptyset$ and the whole universe X are $\#$–open sets.

Proposition 3. *Let* $\langle \mathcal{P}(X), \cap, \cup, {}^c, \emptyset, X\rangle$ *be the algebraic structure generated by the preclusivity space* $\langle X, \#\rangle$ *and* $^{\#}$ *the preclusive complement on* $\mathcal{P}(X)$*. Then, the mapping*

$$\mathbf{U}_{\#} : \mathcal{P}(X) \to \mathcal{P}(X), \quad H \to \mathbf{U}_{\#}(H) := H^{\#\#}$$

is a closure operator*, i.e.,*

$$\begin{array}{lll} (C_0) & \emptyset = \mathbf{U}_{\#}(\emptyset) & (\mathit{normalized}) \\ (C_1) & H \subseteq \mathbf{U}_{\#}(H) & (\mathit{increasing}) \\ (C_2) & \mathbf{U}_{\#}(H) = \mathbf{U}_{\#}(\mathbf{U}_{\#}(H)) & (\mathit{idempotent}) \\ (C_3) & \mathbf{U}_{\#}(H) \cup \mathbf{U}_{\#}(K) \subseteq \mathbf{U}_{\#}(H \cup K) & (\mathit{sub} - \mathit{additive}) \end{array}$$

Let us note that condition (C_3), under $(C_0) - (C_1)$, is equivalent to the *monotonicity condition*: $H \subseteq K$ implies $\mathbf{U}_{\#}(H) \subseteq \mathbf{U}_{\#}(K)$. In general, the Kuratowski property of topological closure, $\mathbf{U}_{\#}(H) \cup \mathbf{U}_{\#}(K) = \mathbf{U}_{\#}(H \cup K)$, does not hold.

Example 8. Let us consider the two sets of flats $H = \{f_2\}$ and $K = \{f_5\}$, relative to Table 2. Then, the closure of H, K and their union, relative to $D = \mathit{Att}(X)$, are respectively $\mathbf{U}_{\#}(H) = \{f_1, f_2, f_3\}$, $\mathbf{U}_{\#}(K) = \{f_4, f_5\}$ and $\mathbf{U}_{\#}(\{f_2, f_5\}) = X$. So, $\mathbf{U}_{\#}(H \cup K) = X \neq \{f_1, f_2, f_3, f_4, f_5\} = \mathbf{U}_{\#}(H) \cup \mathbf{U}_{\#}(K)$.

Since according to (C_1) one has that $H \subseteq \mathbf{U}_{\#}(H)$, it is possible to introduce the collection of all #–*closed* sets defined as follows:

$$\mathbb{C}(X, \#) := \{B \subseteq X : B = \mathbf{U}_{\#}(B) = B^{\#\#}\}.$$

Also this set is not empty since the empty set $\emptyset$ and the whole universe X are both #–closed. It is easy to see that A is #–open iff A^c is #–closed, and similarly B is #–closed iff B^c is #–open. Hence,

$$\mathbb{O}(X, \#) = \{A \subseteq X : A^c \in \mathbb{C}(X, \#)\}$$
$$\mathbb{C}(X, \#) = \{B \subseteq X : B^c \in \mathbb{O}(X, \#)\}.$$

If a set is both #–open and #–closed, it is said to be #–*clopen*. So, the collection of all #–clopen sets is defined as:

$$\mathbb{CO}(X, \#) = \mathbb{C}(X, \#) \cap \mathbb{O}(X, \#).$$

Both the empty set $\emptyset$ and the whole universe X are #–clopen. In the sequel, if there is no confusion, we simply speak of open, closed, and clopen sets instead of #–open, #–closed, and #–clopen sets.

By the increasing (resp., decreasing) property (C_1) (resp., (I_1)) of the closure (resp., interior) operator, the following holds:

$$\mathbf{L}_{\#}(H) \subseteq H \subseteq \mathbf{U}_{\#}(H). \tag{5}$$

Therefore, the pair $\langle \mathbf{L}_{\#}(H), \mathbf{U}_{\#}(H)\rangle$ can be thought of as a preclusive rough approximation of the set H by a preclusive open–closed pair. In fact,

1. by (C_2) we have that $\mathbf{U}_\#(H)$ is a closed set, and inclusion $H \subseteq \mathbf{U}_\#(H)$ says that $\mathbf{U}_\#(H)$ is one of the possible upper closed approximations of H;
2. for every closed set B which is an upper approximation of H, $H \subseteq B$, we have by monotonicity that $\mathbf{U}_\#(H) \subseteq \mathbf{U}_\#(B) = B$,
3. thus, $\mathbf{U}_\#(H)$ is the best approximation of H from the top by closed elements (the *upper closed approximation* of H).

By the above properties (1)–(3), it follows that the preclusive upper approximation of a set H can also be expressed as:

$$\mathbf{U}_\#(H) = \cap\{B \in \mathbb{C}(X, \#) : \; H \subseteq B\}. \tag{6}$$

In the case of a closed set H one has that $H = \mathbf{U}_\#(H)$, i.e., the upper closed approximation of any closed set is the set itself. In this sense we can say that closed sets are *upper exact sets.*

Analogously, for the lower preclusive approximation:

1. by (I_2) we have that $\mathbf{L}_\#(H)$ is an open set, and inclusion $\mathbf{L}_\#(H) \subseteq H$ says that $\mathbf{L}_\#(H)$ is one of the possible lower open approximations of H;
2. for every open set B which is a lower approximation of H, $B \subseteq H$, we have by monotonicity that $B = \mathbf{L}_\#(B) \subseteq \mathbf{L}_\#(H)$,
3. thus, $\mathbf{L}_\#(H)$ is the best approximation of H from the bottom by open elements (the *lower open approximation* of H).

By the above properties (1)–(3), it follows that the preclusive lower approximation of a set H can be expressed as:

$$\mathbf{L}_\#(H) = \cup\{B \in \mathbb{C}(X, \#) : \; B \subseteq H\}. \tag{7}$$

In the case of an open set H obviously $H = \mathbf{L}_\#(H)$, obtaining that open sets can be considered as *lower exact sets.*

Finally, we have that clopen sets are both lower and upper exact sets, so, we simply call them *exact sets.*

Example 9. Making reference to the information system described by Table 2, let us consider the set $H = \{f_4, f_5, f_6\}$ and compute its preclusive rough approximation with respect to the preclusive relation (4) induced by the set $\mathcal{D} = \mathit{Att}(X)$ of all involved attributes. We have that $H^\# = \{f_1, f_3\}$ and $\mathbf{U}_\#(H) = H^{\#\#} = \{f_4, f_5, f_6\} = H$. Computing the preclusive lower approximation, one obtains that $H^c = \{f_1, f_2, f_3\}$, $H^{c\#} = \{f_4, f_5\}$, $H^{c\#\#} = \{f_1, f_2, f_3\}$ and finally, $H^{c\#\#c} = \mathbf{L}_\#(H) = \{f_4, f_5, f_6\} = H$. Thus, the rough approximation of $H = \{f_4, f_5, f_6\}$ is the pair $\langle\{f_4, f_5, f_6\}, \{f_4, f_5, f_6\}\rangle$, and H is a clopen set.

In Figures 1–3, the closed, open, and clopen sets relative to the preclusive relation induced from Table 2 by the set $\mathcal{D} = \mathit{Att}(X)$ are represented through Hasse diagrams.

In Table 3 examples of similar and preclusive approximations with respect to some particular subsets of the involved universe are reported.

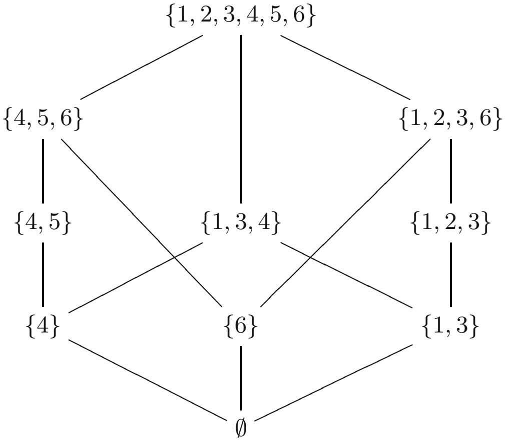


Fig. 1. Closed sets relative to Table 2 for $\mathcal{D} = Att(x)$.

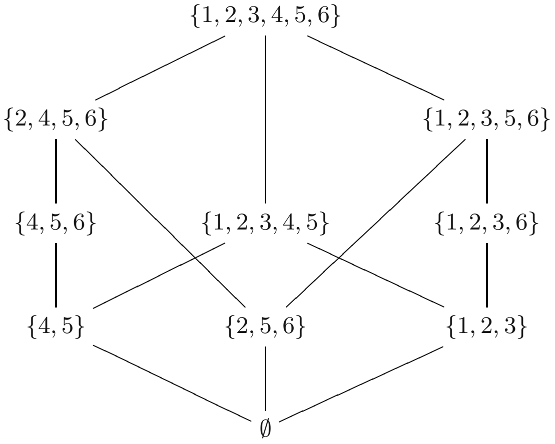


Fig. 2. Open sets relative to Table 2 for $\mathcal{D} = Att(x)$.

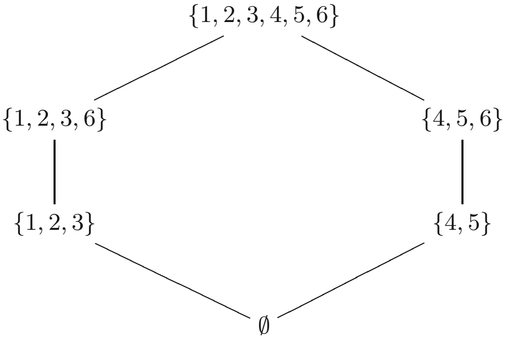


Fig. 3. Clopen sets relative to Table 2 for $\mathcal{D} = Att(x)$.

Table 3. Examples of Approximations.

$\mathbf{L}_{\mathcal{R}}(H)$	$\mathbf{L}_{\#}(H)$	H	$\mathbf{U}_{\#}(H)$	$\mathbf{U}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	$\{f_2\}$	$\{f_1, f_2, f_3\}$	$\{f_1, f_2, f_3, f_6\}$
$\{f_4\}$	$\{f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_4, f_5, f_6\}$
$\{f_4\}$	$\{f_4, f_5\}$	$\{f_1, f_2, f_4, f_5\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
$\{f_6\}$	$\{f_2, f_5, f_6\}$	$\{f_1, f_2, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$

As can be seen, in all these particular cases the following chain of inclusions holds:

$$\mathbf{L}_{\mathcal{R}}(H) \subseteq \mathbf{L}_{\#}(H) \subseteq H \subseteq \mathbf{U}_{\#}(H) \subseteq \mathbf{U}_{\mathcal{R}}(H) \tag{8}$$

Thus, it seems possible to conjecture that the preclusive approximation gives a better, i.e., closer, approximation of a given set H than the corresponding similarity one. We will show in Section 3 that this is a general property, which holds for all subsets of the universe of a preclusivity space.

2 About the Monotone Increase of Knowledge in Time

Starting from an incomplete information system one can wonder what happens when we have more knowledge about it, i.e., when the number of unknown values decreases. One could expect that for a fixed set of attributes to more knowledge there corresponds more open and closed sets producing in this way a preclusive rough approximation which is better than the previous one (if the information increases, then the approximation should be better and better). However, this is not always the case as the following examples will show.

Example 10. We now consider two examples showing two different cases making reference to the information system of Table 2 which is supposed relative to a given time, say t_1. Let us consider a first situation, which can be attributed to a knowledge situation of some observer O_1 at a time, say t_0, former than the time t_1 (i.e., $t_0 < t_1$) in which the knowledge of Table 2 has been achieved. This first case corresponds to a worst knowledge about flat f_3 according to the situation described in Table 4.

Table 4. Case 1: only the information regarding flat f_3 is different from the one of Table 2 with a global increase in time of the knowledge.

Flat	Price	Rooms	Down-Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	*	no
f_4	low	*	no	no
f_5	low	1	*	no
f_6	*	1	yes	*

In the second case, another observer O_2 is involved whose former knowledge, relative to time t_0, with respect to the information system of Table 2 consists of a worst knowledge about flat f_6 according to the situation described in Table 5.

Table 5. Case 2: only the information regarding flat f_6 is different from the one of Table 2 with a global increase in time of the knowledge.

Flat	Price	Rooms	Down-Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	yes	no
f_4	low	*	no	no
f_5	low	1	*	no
f_6	*	1	*	*

Thus, there are two observers that initially (time t_0) have a different knowledge about the same collection of flats (described by Tables 4 and 5). During the time, both the observers increase their knowledge reaching the same result exposed in Table 2 at a subsequent time t_1. The final result is that relatively to the same set of all attributes $\mathcal{D} = Att(X)$ there is a decrease in the number of closed (and so also of open) sets passing from case 1 (time t_0) to the situation of Table 2 (time t_1) and an increase going from case 2 (time t_0) to Table 2 (time t_1), as can be seen looking at Figures 4 and 5.

When considering the clopen sets we observe that their number increases in the original situation depicted in figure 3 with respect to both case 1 and case 2, as can be seen in Figure 6.

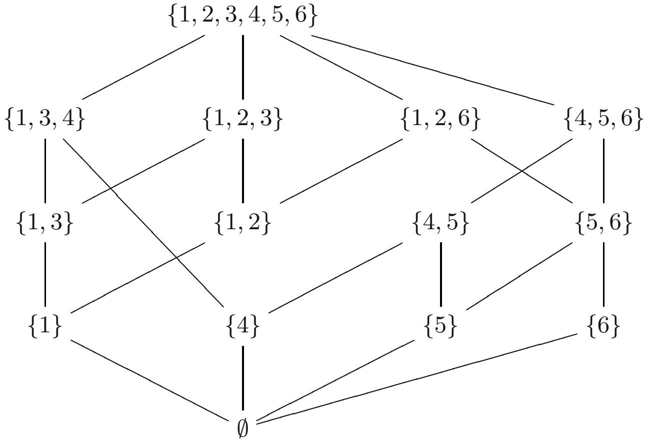


Fig. 4. Closed sets in case 1 at time t_0: during the time evolution from t_0 to t_1 (corresponding to Figure 1) there is a decrease of the total number of closed sets with the appearance of the set $\{1, 2, 3, 6\}$.

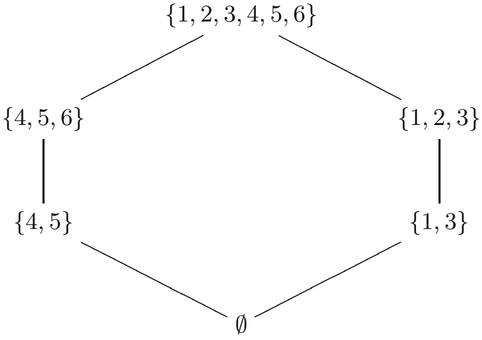


Fig. 5. Closed sets in case 2 at time t_0: during the time evolution from t_0 to t_1 (corresponding to Figure 1) there is an increase of the total number of closed sets.

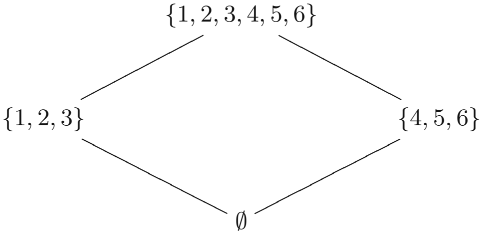


Fig. 6. Clopen Sets: case 1 and case 2 at time t_0.

Again we ask whether this is a general property: to a greater knowledge corresponds a higher number of clopen sets. Also in this case, the answer is negative. Let us suppose that, with respect to the original situation of Table 2 at time t_1, both the observers in a later time, say t_2, increase their knowledge about flat 5 according to Table 6.

Table 6. Case 3 at time $t_2 > t_1$.

Flat	Price	Rooms	Down-Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	yes	no
f_4	low	*	no	no
f_5	low	1	no	no
f_6	*	1	yes	*

In this case, however, the number of clopen sets decreases with respect to the knowledge at time t_1, as can be seen in Figure 7.

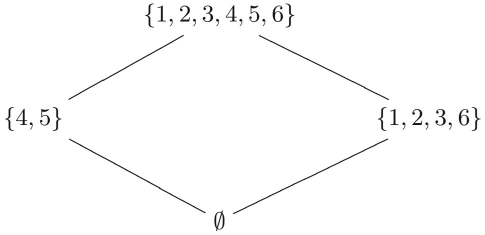


Fig. 7. Clopen Sets: Case 3 at time $t_2 > t_1$.

When considering the closed sets, it happens that they are numerically less at time t_2 with respect to the ones at time t_1, but the set $\{1,3,4,5\}$ is closed at time t_2 and not at time t_1.

In Figure 8 it is reported the Hasse diagram of closed sets at time t_2.

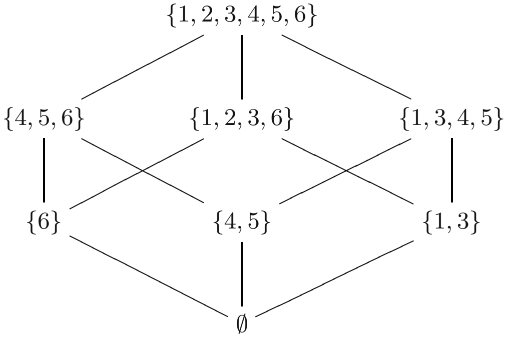


Fig. 8. Closed sets: Case 3 at time $t_2 > t_1$.

Finally, as a limit final case we consider the complete information system of Table 1, corresponding to an increase of the knowledge represented by Table 6 (occurred at a time $t_3 > t_2$). In this case the set of closed and clopen sets are the same as the ones of case 3 (see Figures 8 and 7 respectively).

We can conclude that there is no correspondence between unknown values and exact (either open or closed or clopen sets) of the preclusive environment. The same uncertainty holds with respect to the quality of an approximation. Let us consider an incomplete information system based on the universe X with associated a pair $(\mathcal{R}, \#)$ of correlated similarity–preclusive relations. For a given set of objects H from X we can compute its preclusive rough approximation. Now, if we add (resp., remove) some information to (resp., from) the system, and recompute the preclusive approximation, we have no idea if the result is better or worst than the previous one.

Example 11. Let us consider the information system at time t_0 of Table 4. If we compute the approximations of the same sets used in Table 3, we obtain the following results.

Table 7. Examples of Approximations, case 1 at time t_0.

$\mathbf{L}_{\mathcal{R}}(H)$	$\mathbf{L}_{\#}(H)$	H	$\mathbf{U}_{\#}(H)$	$\mathbf{U}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	$\{f_2\}$	$\{f_1, f_2\}$	$\{f_1, f_2, f_3, f_6\}$
$\emptyset$	$\emptyset$	$\{f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_3, f_4, f_5, f_6\}$
$\emptyset$	$\emptyset$	$\{f_1, f_2, f_4, f_5\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
$\{f_6\}$	$\{f_2, f_5, f_6\}$	$\{f_1, f_2, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$

That is, going from a situation at time t_0 to a situation at time t_1, i.e., adding knowledge to the information system in a monotonic way, we have that

1. the time evolution of the preclusive rough approximation of a set is unpredictable, i.e., the approximation becomes either worst (case $\{f_2\}$), or better (cases $\{f_4, f_5\}$, $\{f_1, f_2, f_4, f_5\}$), or remains the same (case $\{f_1, f_2, f_5, f_6\}$);
2. on the contrary, in this example the similarity rough approximation at time t_1 of a set is always better than the corresponding similarity rough approximation of the same set at the previous time t_0, i.e., there is a *monotone* behavior of the similarity approximation with respect to the knowledge increase;
3. however, at any fixed time the preclusive approximation of a set is always better then the correlated similarity approximation, i.e., the chain (8) holds for any set.

As regards to the preclusive approximations, we can select those situations in which to an increase of knowledge in time corresponds an increase in the quality of the approximations.

First of all, let us introduce the following definitions.

Definition 5. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$, with $t_0, t_1 \in \mathbb{R}$, be two incomplete information systems based on the same triple $\langle X, Att(X), val(X)\rangle$ and characterized by two different information maps*

$$F^{(t_0)} : X \times Att(X) \mapsto val(X) \quad \textit{and} \quad F^{(t_1)} : X \times Att(X) \mapsto val(X).$$

We will say that there is a monotonic increase of information *iff*

$$t_0 \leq t_1 \quad \textit{and} \quad \forall (x, a)\ (F^{(t_0)}(x, a) \neq * \ \Rightarrow\ F^{(t_1)}(x, a) = F^{(t_0)}(x, a)).$$

In such a case, we will write $\mathcal{K}^{(t_0)}(X) \preceq \mathcal{K}^{(t_1)}(X)$.

Definition 6. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two incomplete information systems such that $\mathcal{K}^{(t_0)}(X) \preceq \mathcal{K}^{(t_1)}(X)$. We will say that there is a* monotonic increase of knowledge *iff $\mathbb{C}^{(t_0)}(X) \subseteq \mathbb{C}^{(t_1)}(X)$.*

In the previous examples we have seen that to a monotonic increase of information does not correspond a monotonic increase in the quality of the approximations. This desirable behavior holds, instead, in the case of monotonic increase of knowledge.

Proposition 4. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two incomplete information systems characterized by a monotonic increase of knowledge. Then*

$$\forall H \subseteq X \quad \mathbf{L}_{\#}^{(t_0)}(H) \subseteq \mathbf{L}_{\#}^{(t_1)}(H) \subseteq H \subseteq \mathbf{U}_{\#}^{(t_1)}(H) \subseteq \mathbf{U}_{\#}^{(t_0)}(H). \tag{9}$$

Proof. By hypothesis, $\mathbb{C}^{(t_0)}(X) \subseteq \mathbb{C}^{(t_1)}(X)$ and consequently $\mathbb{O}^{(t_0)}(X) \subseteq \mathbb{O}^{(t_1)}(X)$. Now, let us consider the rough lower approximation of a set H, as defined in equation (7). Suppose that there exists a set $A \in \mathbb{O}^{(t_0)}(X)$ such that $A \subseteq H$, then, by hypothesis, $A \in \mathbb{O}^{(t_1)}(X)$. So $\mathbf{L}_{\#}^{(t_0)}(H) = \cup\{A \in \mathbb{O}^{(t_0)}(X) : A \subseteq H\} \subseteq \cup\{A \in \mathbb{O}^{(t_1)}(X) : A \subseteq H\} = \mathbf{L}_{\#}^{(t_1)}(H)$. On the other hand, suppose that there exists $B \in \mathbb{C}^{(t_0)}(X)$ such that $H \subseteq B$, then it also holds $B \in \mathbb{C}^{(t_1)}(X)$. So, $\mathbf{U}_{\#}^{(t_1)}(H) = \cap\{B \in \mathbb{C}^{(t_1)}(X) : H \subseteq B\} \subseteq \cap\{B \in \mathbb{C}^{(t_0)}(X) : H \subseteq B\} = \mathbf{U}_{\#}^{(t_0)}(H)$

Example 12. Let us consider the case 2, at time t_0, and compute the approximations of the same sets of Table 3. The results are shown in Table 8.

Table 8. Examples of Approximations, case 2.

$\mathbf{L}_{\mathcal{R}}(H)$	$\mathbf{L}_{\#}(H)$	H	$\mathbf{U}_{\#}(H)$	$\mathbf{U}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	$\{f_2\}$	$\{f_1, f_2, f_3\}$	$\{f_1, f_2, f_3, f_6\}$
$\emptyset$	$\emptyset$	$\{f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_4, f_5, f_6\}$
$\emptyset$	$\emptyset$	$\{f_1, f_2, f_4, f_5\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
$\emptyset$	$\emptyset$	$\{f_1, f_2, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$

Going from Table 5 (case 2), at time t_0, to Table 2, at time t_1, we have a monotonic increase of knowledge. So, as expected, it results that the approximations computed in Table 3 are always better or equal to the ones of Table 8.

Differently from the preclusive rough approximation, if we consider the similarity rough approximation, we can see, comparing Tables 7 and 8 with Table 3, that the quality of the approximation is monotonic with respect to the quantity of information. This is a general result, as shown in the following proposition.

Proposition 5. *Let $\mathcal{K}^{(t_0)}(X)$ and $\mathcal{K}^{(t_1)}(X)$ be two incomplete information systems such that $\mathcal{K}^{(t_0)}(X) \preceq \mathcal{K}^{(t_1)}(X)$ (there is a monotonic increase of information). Then,*

$$\forall H \subseteq X \quad \mathbf{L}_{\mathcal{R}}^{(t_0)}(H) \subseteq \mathbf{L}_{\mathcal{R}}^{(t_1)}(H) \subseteq H \subseteq \mathbf{U}_{\mathcal{R}}^{(t_1)}(H) \subseteq \mathbf{U}_{\mathcal{R}}^{(t_0)}(H). \tag{10}$$

Proof. Considering the similarity relation defined in equation (1), we have that if two objects are similar at time t_1 then they are also similar at time t_0: $x\mathcal{R}^{(t_1)}y$

implies $x\mathcal{R}^{(t_0)}y$. So the granule (similarity class) of an object x at time t_0 is bigger than the the granule (similarity class) of x at time t_1: $S^{(t_1)}(x) \subseteq S^{(t_0)}(x)$. Now, let H be a set of objects and compute its similarity rough approximation as defined in Equations (2a) and (2b). As regards to the lower approximation, we have that if $x \in L_{\mathcal{R}}^{(t_0)}(H)$, then $S^{(t_0)}(x) \subseteq H$. So, $S^{(t_1)}(x) \subseteq S^{(t_0)}(x) \subseteq H$, that is $x \in L_{\mathcal{R}}^{(t_1)}(H)$, and finally, $L_{\mathcal{R}}^{(t_0)}(H) \subseteq L_{\mathcal{R}}^{(t_1)}(H)$. Similarly, for the upper approximation, if $x \in U_{\mathcal{R}}^{(t_1)}(H)$, one obtains by definition that $S^{(t_1)}(x) \cap H \neq \emptyset$. Then $S^{(t_0)}(x) \cap H \neq \emptyset$, i.e., $x \in U_{\mathcal{R}}^{(t_0)}(H)$ and $U_{\mathcal{R}}^{(t_1)}(H) \subseteq U_{\mathcal{R}}^{(t_0)}(H)$.

Concluding, if we suppose an increase of the information in time we have a monotone behaviour of the similarity approximation and an unpredictable behaviour of the preclusive approximation, with the last one which is always better than the first one. From an intuitive point of view we can imagine a situation similar to the one drawn in Figure 9.

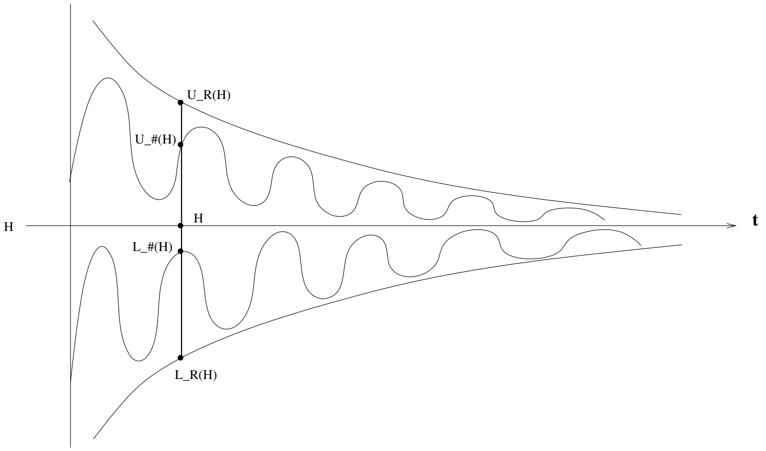


Fig. 9. An imaginary representation of the time evolution of the similarity and preclusive rough approximations.

All the examples considered until now are about incomplete information systems and the similarity relation given by Equation (1). However, the pathological behavior remarked about the monotonic increase of information holds also in other contexts.

Another typical case of a reflexive and symmetric binary relation which in general is not transitive is the one induced by any distance. Precisely, if it can be introduced a metric assigning a distance between pairs of objects of the universe of an information system, we could say that two objects are similar if their distance is less than a fixed value ϵ. In this situation the similarity relation is sometimes called ϵ-*indiscernibility* and “the basic idea is that small differences on some attribute values may be judged insignificant” [SV95].

2.1 ϵ-Indiscernibility in Presence of a Metric

Let $\mathcal{K}$ be an information system with numeric valued attributes, i.e., $Att(X) \subseteq \mathbb{R}$. Then, given an attribute $a \in Att(X)$ a way to define a pseudo–metric for a is

$$d_a(x,y) := \frac{|F(x,a) - F(y,a)|}{\max\{F(z,a) : z \in X\} - \min\{F(w,a) : w \in X\}} \quad . \tag{11}$$

Clearly, this is a pseudo–metric and not a metric since it is not true that $d_a(x,y) = 0$ implies $x = y$ [Kel55]. Further, we remark that it is well defined since $\max\{F(z,a) : z \in X\} - \min\{F(w,a) : w \in X\}$ is never equal to 0 due to the assumption that for all attributes a there exist at least to objects x_1, x_2 such that $F(x_1,a) \neq F(x_2,a)$ (see Section 1).

In presence of two or more attributes collected in a set $\mathcal{D} \subseteq Att(X)$, we can introduce an overall distance:

$$d_D(x,y) := \sum_{a \in D} \omega_a d_a(x,y),$$

where ω_a are weights such that $\sum \omega_a = 1$ (a standard choice is $\omega_a = \frac{1}{|D|}$). So, for a fixed value $\epsilon \in [0,1]$ we can consider the similarity relation defined as follows:

$$x \mathcal{R}_D^{\epsilon} y \quad \text{iff} \quad d_D(x,y) \leq \epsilon. \tag{12}$$

The correlated preclusive relation is the following:

$$x \#_D^{\epsilon} y \quad \text{iff} \quad d_D(x,y) > \epsilon.$$

Let us stress that this is not the unique possibility to define a metric, for a more complete discussion see [NSS98].

Example 13. We apply now the similarity relation (12) on the same example used in [SV95] and [GMS98]. Let us consider the situation of 12 firms described in Table 9.

These firms are analyzed with respect to the following attributes:

- a_1= value of fixed capital,
- a_2= value of sales in the year preceding the application,
- a_3= kind of activity, classified according to the typology $0, 1$ and 2,

Further, these firms are divided in two classes: paid back their credit (d=1) or not (d=2).

In order to have a similarity relation, we define a distance between two objects by considering only the first two attributes, i.e., $D = \{a_1, a_2\}$. So, by equation (12), the family of similarity relations, depending on ϵ, is:

$$x \mathcal{R}_D^{\epsilon} y \quad \text{iff} \quad \frac{1}{2}\Big(\frac{|F(x,a_1) - F(y,a_1)|}{91} + \frac{|F(x,a_2) - F(y,a_2)|}{53} \Big) \leq \epsilon.$$

Table 9. Firms information table.

Firm	a_1	a_2	a_3	d
1	43	78	0	1
2	54	75	0	2
3	124	50	1	1
4	102	65	1	2
5	98	80	2	2
6	88	102	2	2
7	130	57	0	1
8	128	92	1	2
9	82	59	1	1
10	134	103	2	2
11	58	55	0	1
12	126	71	1	2

Let us set $\epsilon = 0.4$. We have, for example, that firm 1 is similar to firm 5:

$$d_D(1,5) = \frac{1}{2}\Big(\frac{|43-98|}{91} + \frac{|78-80|}{53}\Big) \simeq 0.32 < 0.4 = \epsilon$$

On the other hand, firm 1 is preclusive to firm 12:

$$d_D(1,12) = \frac{1}{2}\Big(\frac{|43-126|}{91} + \frac{|78-71|}{53}\Big) \simeq 0.52 > 0.4 = \epsilon$$

Further, firm 5 is similar to firm 12:

$$d_D(5,12) = \frac{1}{2}\Big(\frac{|98-126|}{91} + \frac{|71-80|}{53}\Big) \simeq 0.24 < 0.4 = \epsilon$$

Thus, it is easily verified that $\mathcal{R}_D^{0.4}$ is not a transitive relation, since firm 1 is similar to firm 5 which is similar to firm 12, but firm 1 and firm 12 are not similar.

Now, let us consider the subsets of firms which paid back and did not pay back their credit: $P = \{1,3,7,9,11\}$ and $N = \{2,4,5,6,8,10,12\}$. In Table 10 we report their similarity and preclusive rough approximations, with respect to the relation $\mathcal{R}^{0.2}$.

Table 10. Example of rough approximations of the sets P and N.

$\mathbf{L}_{\mathcal{R}}(H)$	$\mathbf{L}_{\#}(H)$	H	$\mathbf{U}_{\#}(H)$	$\mathbf{U}_{\mathcal{R}}(H)$
{3,11}	{3,7,9,11}	P	{1,2,3,7,9,11,12}	{1,2,3,4,7,9,11,12}
{5,6,8,10}	{4,5,6,8,10}	N	{1,2,4,5,6,8,10,12}	{1,2,4,5,6,7,8,9,10,12}

As expected, preclusive approximations are better than the corresponding similarity ones. This is particularly evident when considering the boundary region, i.e., $\mathbf{U}(H)/\mathbf{L}(H)$, which represents the elements whose classification is uncertain. In fact, considering the set P we have that its preclusive boundary region $\mathbf{U}_{\#}(P)/\mathbf{L}_{\#}(P)$ is the set $\{1,2,12\}$, whereas its similarity boundary region

$\mathbf{U}_{\mathcal{R}}(P)/\mathbf{L}_{\mathcal{R}}(P)$ is the set $\{1, 2, 4, 7, 9, 12\}$. Hence, the number of elements whose belonging to P is uncertain is half in the case of preclusive with respect to the corresponding similarity approximation.

Now, we turn our attention to open and closed sets. As can be seen in Table 11, the number of exact sets, both closed (and so also open) and clopen, decreases as the value of ϵ increases.

Table 11. Number of exact sets, depending on ϵ.

ϵ	Closed	Clopen
0.1	1024	1024
0.2	192	80
0.3	74	20
0.4	34	4
0.5	26	2
0.6	16	2
0.7	8	2
0.8	4	2
0.9	2	2

Thus, one could think that given $\epsilon_1 \leq \epsilon_2$, if H is closed with respect to $\#^{\epsilon_2}$, i.e., $H = \mathbf{U}^{\epsilon_2}_{\#}(H)$, then H is closed also with respect to $\#^{\epsilon_1}$, i.e., $H = \mathbf{U}^{\epsilon_1}_{\#}(H)$. In other words, one expects that $\epsilon_1 \leq \epsilon_2$ implies $\mathbb{C}^{\epsilon_2}(X) \subseteq \mathbb{C}^{\epsilon_1}(X)$. However, this is not true. For example, we have that $\{8, 10\} \in \mathbb{C}^{0.7}(X)$ but $\{8, 10\} \notin \mathbb{C}^{0.6}(X)$. The same can be said about clopen sets, in fact, we have that $\{5, 7, 8, 10\} \in \mathbb{CO}^{0.3}(X)$ but $\{5, 7, 8, 10\} \notin \mathbb{CO}^{0.2}(X)$.

When considering the rough approximation of a set H with respect to different relations relative to $\epsilon_1 \leq \epsilon_2$, we have the same non–predictability encountered in the case of incomplete information systems at different time steps $t_1 \leq t_2$. Indeed, we can have that the approximation relative to ϵ_2 is better, worst or equal to the approximation relative to $\epsilon_1 \leq \epsilon_2$.

Let us refer to the information system of Table 9. We set $D = \{a_1, a_2\}$ and compute the approximation of some sets using the relation (12) with respect to two different values $\epsilon_1 = 0.6$ and $\epsilon_2 = 0.7$. The results are reported in Tables 12 and 13, respectively.

Table 12. Example of approximation for $\epsilon_1 = 0.6$.

$\mathbf{L}^{\epsilon_1}_{\mathcal{R}}(H)$	$\mathbf{L}^{\epsilon_1}_{\#}(H)$	H	$\mathbf{U}^{\epsilon_1}_{\#}(H)$	$\mathbf{U}^{\epsilon_1}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	{8,10}	{6,8,10}	{1,2,3,4,5,6,7,8,9,10,12}
$\emptyset$	$\emptyset$	{2}	{1,2}	{1,2,4,5,6,7,8,9,11,12}
$\emptyset$	$\emptyset$	{2,4,5}	X	X

Table 13. Example of approximation for $\epsilon_2 = 0.7$.

$\mathbf{L}^{\epsilon_2}_{\mathcal{R}}(H)$	$\mathbf{L}^{\epsilon_2}_{\#}(H)$	H	$\mathbf{U}^{\epsilon_2}_{\#}(H)$	$\mathbf{U}^{\epsilon_2}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	{8,10}	{8,10}	{1,2,3,4,5,6,7,8,9,10,12}
$\emptyset$	$\emptyset$	{2}	{1,2,9,11}	{1,2,3,4,5,6,7,8,9,11,12}
$\emptyset$	$\emptyset$	{2,4,5}	X	X

Thus, in the considered three cases we have all the possible situations: the preclusive approximation relative to ϵ_1 is better (case $\{2\}$), worse (case $\{8, 10\}$) or equal (case $\{2, 4, 5\}$) to the approximation relative to ϵ_2.

A particular situation that guarantees us to have a better approximation going from ϵ_1 to ϵ_2 is given by a monotonic (with respect to ϵ) increase of knowledge.

Proposition 6. *Let $\mathcal{K}(X)$ be a real valued information system (i.e., $val(X) \subseteq \mathbb{R}$) and let $\epsilon_1, \epsilon_2 \in [0, 1]$. If, with respect to the similarity relation* (12)*, it happens that $\mathbb{C}^{\epsilon_2}(X) \subseteq \mathbb{C}^{\epsilon_1}(X)$ then*

$$\forall H \subseteq X \quad \mathbf{L}^{\epsilon_2}_{\#}(H) \subseteq \mathbf{L}^{\epsilon_1}_{\#}(H) \subseteq H \subseteq \mathbf{U}^{\epsilon_1}_{\#}(H) \subseteq \mathbf{U}^{\epsilon_2}_{\#}(H).$$

Proof. Same as Proposition 4.

Example 14. Let us refer to the Example 13. If we choose $\epsilon_1 = 0.7$ and $\epsilon_2 = 0.8$ we have that $\mathbb{C}^{\epsilon_2}(X) \subseteq \mathbb{C}^{\epsilon_1}(X)$, as can be seen by Hasse diagrams of figure 10 and 11.

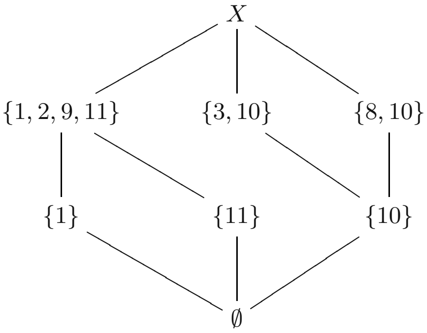


Fig. 10. Closed sets: case $\epsilon_1 = 0.7$.

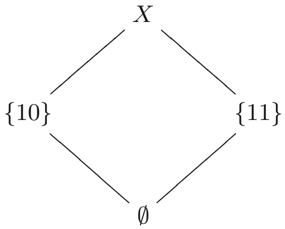


Fig. 11. Closed sets: case $\epsilon_2 = 0.8$.

So we expect that the approximations relative to $\epsilon_1 = 0.7$ are better than the ones relative to $\epsilon_2 = 0.8$. As an example we compute the approximations of the same sets of Example 13 in the case of $\epsilon_2 = 0.8$.

Table 14. Example of approximation, $\epsilon = 0.8$.

$\mathbf{L}^{\epsilon}_{\mathcal{R}}(H)$	$\mathbf{L}^{\epsilon}_{\#}(H)$	H	$\mathbf{U}^{\epsilon}_{\#}(H)$	$\mathbf{U}^{\epsilon}_{\mathcal{R}}(H)$
$\emptyset$	$\emptyset$	{8,10}	X	X
$\emptyset$	$\emptyset$	{2}	X	X
$\emptyset$	$\emptyset$	{2,4,5}	X	X

By comparing Table 13 with Table 14, it is easily seen that Proposition 6 is verified.

On the other hand, in the case of the similarity rough approximation, we always have a monotonic behavior.

Proposition 7. *Let $\mathcal{K}(X)$ be a real valued information system (i.e., $val(X) \subseteq \mathbb{R}$) and let $\epsilon_1, \epsilon_2 \in [0,1]$ with $\epsilon_1 \leq \epsilon_2$. Then, once a set of attributes $D \subseteq Att(X)$ is chosen and the similarity relation $\mathcal{R}^{\epsilon}_{D}$ is defined as in equation* (12)*, the following holds*

$$\mathbf{L}^{\epsilon_2}_{\mathcal{R}}(H) \subseteq \mathbf{L}^{\epsilon_1}_{\mathcal{R}}(H) \subseteq H \subseteq \mathbf{U}^{\epsilon_1}_{\mathcal{R}}(H) \subseteq \mathbf{U}^{\epsilon_2}_{\mathcal{R}}(H).$$

Proof. The proof is similar to the one of proposition 5, considering that if $\epsilon_1 \leq \epsilon_2$ than $x\mathcal{R}^{\epsilon_1}y$ implies $x\mathcal{R}^{\epsilon_2}y$.

3 Quasi-BZ Distributive Lattices

In the previous sections, through concrete cases based on the power set of a universe, we introduced preclusivity spaces and the rough approximations generated by them. Moreover, it emerged that in all the considered examples the preclusive approximation was always better than the corresponding similarity approximation. The aim of this section is to give an algebraic characterization to preclusivity spaces in terms of quasi–Brouwer–Zadeh distributive lattice structures ([Cat97,Cat98]). This will also lead to a theoretical proof of the better behavior of the preclusive rough approximation with respect to the similarity one.

In particular, we considered the concrete algebra $\langle \mathcal{P}(X), \cup, \cap, {}^{c}, {}^{\#} \rangle$ based on the power set $\mathcal{P}(X)$ of a universe X endowed with the set theoretic union, intersection, complement operations, and an unusual complementation ${}^{\#}$.

Now, through an abstraction, we are going to introduce an abstract system consisting of a set Σ, corresponding to the power set $\mathcal{P}(X)$ of the concrete example, upon which the following operations are defined:

- two binary operations, $\vee$ and $\wedge$, corresponding respectively to the set theoretic union and intersection of the concrete example;
- two unary operations, $'$ and ${}^{\sim}$, corresponding to the standard and preclusive complementations of the concrete example.

Definition 7. *A system $\langle \Sigma, \wedge, \vee, ', {}^{\sim}, 0, 1 \rangle$ is a* quasi–Brouwer–Zadeh (BZ) distributive lattice *if the following properties hold:*

(1) *Σ is a distributive lattice with respect to the join and the meet operations $\vee, \wedge$ whose induced partial order relation is*

$$a \leq b \quad \textit{iff} \quad a = a \wedge b \quad (\textit{equivalently}\, b = a \vee b)$$

Moreover, it is required that Σ is bounded by the least element 0 and the greatest element 1:

$$\forall a \in \Sigma,\ 0 \leq a \leq 1$$

(2) *The unary operation $' : \Sigma \mapsto \Sigma$ is a* Kleene *(also* Zadeh *or* fuzzy*) complementation. In other words for arbitrary $a, b \in \Sigma$:*

(K1) $a'' = a$
(K2) $(a \vee b)' = a' \wedge b'$
(K3) $a \wedge a' \leq b \vee b'$.

(3) *The unary operation $^\sim : \Sigma \mapsto \Sigma$ is a* Brouwer *(or* intuitionistic*) complementation. In other words for arbitrary $a, b \in \Sigma$:*

(B1) $a \wedge a^{\sim\sim} = a$
(B2) $(a \vee b)^\sim = a^\sim \wedge b^\sim$
(B3) $a \wedge a^\sim = 0$

(4) *The two complementations are linked by the interconnection rule which must hold for arbitrary $a \in \Sigma$:*

(in) $a^\sim \leq a'$

We remark that under condition (K1), property (K2) is equivalent to the following

(K2a) $(a \wedge b)' = a' \vee b'$;
(K2b) $a \leq b$ implies $b' \leq a'$;
(K2c) $b' \leq a'$ implies $a \leq b$.

In general neither the non contradiction law $a \wedge a' = 0$ nor the excluded middle law $a \vee a' = 1$ for the Kleene negation are required to hold.

As to the intuitionistic negation, we have that under condition (B1), condition (B2) is equivalent only to the weak form of non contraposition law

(B2a) $a \leq b$ implies $b^\sim \leq a^\sim$.

In general the strong contraposition law, $b^\sim \leq a^\sim$ implies $a \leq b$, and the dual de Morgan law, $(a \wedge b)^\sim = a^\sim \vee b^\sim$ do not hold for the Brouwer negation. As expressed by property (B3), the non contradiction law is satisfied by all elements, whereas, in general, the excluded middle law is not required to hold.

In the framework of quasi BZ lattices, one can naturally introduce the *anti-Brouwer complement* $^\flat : \Sigma \mapsto \Sigma$ defined for every $a \in \Sigma$ as: $a^\flat := a'^{\sim\prime}$. It can be easily shown that the operation $^\flat$ satisfies the following properties:

(AB1) $a \leq b$ implies $b^\flat \leq a^\flat$ [Contraposition law]
(AB2) $a \vee a^\flat = 1$ [Excluded middle law]
(AB3) $a^{\flat\flat} \leq a$. [Weak double negation law]

Our three complements turn out to be connected by the following chain:

$$\forall a \in \Sigma\ , \quad a^{\sim} \leq a' \leq a^{\flat}.$$

Proposition 8. *Let $\langle \Sigma, \wedge, \vee, ', {}^{\sim}, 0, 1\rangle$ be a quasi BZ distributive lattice. Then, the map $\mathbf{i} : \Sigma \to \Sigma$ such that $\mathbf{i}(a) := a^{\flat\flat} = a'^{\sim\sim\prime}$ is an interior operator. That is, the following are satisfied:*

$$\begin{array}{lll}
(I_0) & 1 = \mathbf{i}(1) & (\textit{normalized}) \\
(I_1) & \mathbf{i}(a) \leq a & (\textit{decreasing}) \\
(I_2) & \mathbf{i}(a) = \mathbf{i}(\mathbf{i}(a)) & (\textit{idempotent}) \\
(I_3) & \mathbf{i}(a \wedge b) \leq \mathbf{i}(a) \wedge \mathbf{i}(b) & (\textit{submultiplicative})
\end{array}$$

Dually, the map $\mathbf{c} : \Sigma \to \Sigma$ such that $\mathbf{c}(a) := a^{\sim\sim}$ is a closure operator, i.e.:

$$\begin{array}{lll}
(C_0) & 0 = \mathbf{c}(0) & (\textit{normalized}) \\
(C_1) & a \leq \mathbf{c}(a) & (\textit{increasing}) \\
(C_2) & \mathbf{c}(a) = \mathbf{c}(\mathbf{c}(a)) & (\textit{idempotent}) \\
(C_3) & \mathbf{c}(a) \vee \mathbf{c}(b) \leq \mathbf{c}(a \vee b) & (\textit{subadditive})
\end{array}$$

Under condition (I_0)–(I_2) the submultiplicative property (I_3) is equivalent to the monotonicity condition $a \leq b$ implies $\mathbf{i}(a) \leq \mathbf{i}(b)$. Dually, under condition (C_0)–(C_2) the subadditive property (C_3) is equivalent to the monotonicity condition $a \leq b$ implies $\mathbf{c}(a) \leq \mathbf{c}(b)$.

Since in general $\mathbf{i}(a) \leq a$, an element $e \in \Sigma$ is said to be *innerdefinable (open)* iff $\mathbf{i}(e) = e$. Analogously, since in general $a \leq \mathbf{c}(a)$, an element $f \in \Sigma$ is said to be *outerdefinable (closed)* iff $f = \mathbf{c}(f)$. We will indicate by $\mathbb{O}(\Sigma)$ the set of all *innerdefinable (open)* elements of the space, while $\mathbb{C}(\Sigma)$ will represent the set of all *outerdefinable (closed)* elements. In other words:

$$\mathbb{O}(\Sigma) = \{e \in \Sigma : e = \mathbf{i}(e)\}$$
$$\mathbb{C}(\Sigma) = \{f \in \Sigma : f = \mathbf{c}(f)\}$$

It is trivial to prove that an element a is closed iff a' is open, and vice versa an element a is open iff a' is closed. In this way there is a duality between open and closed elements determined by the Kleene negation:

$$\mathbb{O}(\Sigma) = \{e \in \Sigma : e' \in \mathbb{C}(\Sigma)\}$$
$$\mathbb{C}(\Sigma) = \{f \in \Sigma : f' \in \mathbb{O}(\Sigma)\}$$

In general, these two subsets of Σ do not coincide, neither one is a subset of the other. Thus, it is worthwhile to consider also the set of all *innouter* (or *clopen*) elements:

$$\mathbb{CO}(\Sigma) := \mathbb{O}(\Sigma) \cap \mathbb{C}(\Sigma).$$

Moreover, $\mathbb{CO}(\Sigma)$ is not empty because it contains the least element 0 and the greatest element 1 of the lattice Σ. In the sequel we are also interested to the so–called *exterior* of an element $a \in \Sigma$ defined as the open element $\mathbf{e}(a) := \mathbf{c}(a)'$.

The above considerations lead to the definition of an abstract approximation space generated by a quasi BZ lattice.

Definition 8. *Let $\langle \Sigma, \leq, ', {}^{\sim}, 0, 1\rangle$ be a quasi BZ distributive lattice. The induced rough approximation space is the structure*

$$\langle \Sigma, \mathbb{O}(\Sigma), \mathbb{C}(\Sigma), \mathbf{i}, \mathbf{c}\rangle,$$

where

- Σ *is the set of approximable elements;*
- $\mathbb{O}(\Sigma) \subseteq \Sigma$ *is the set of* innerdefinable *elements, such that* 0 *and* $1 \in \mathbb{O}(\Sigma)$*;*
- $\mathbb{C}(\Sigma) \subseteq \Sigma$ *is the set of* outerdefinable *elements, such that* 0 *and* $1 \in \mathbb{C}(\Sigma)$*;*
- $\mathbf{i} : \Sigma \to \mathbb{O}(\Sigma)$ *is the* inner approximation map*;*
- $\mathbf{c} : \Sigma \to \mathbb{C}(\Sigma)$ *is the* outer approximation map*;*

For any element $a \in \Sigma$, its rough approximation *is defined as the pair:*

$$r(a) := \langle \mathbf{i}(a), \mathbf{c}(a)\rangle \quad [\textit{with} \quad \mathbf{i}(a) \leq a \leq \mathbf{c}(a)]$$

drawn in the following diagram:

$$\begin{tikzcd}
& a \in \Sigma \arrow[dl, "\mathbf{i}"'] \arrow[dr, "\mathbf{c}"] \arrow[dd, "r"] & \\
\mathbf{i}(a) \in \mathbb{O}(\Sigma) \arrow[dr] & & \mathbf{c}(a) \in \mathbb{C}(\Sigma) \arrow[dl] \\
& \langle \mathbf{i}(a), \mathbf{c}(a)\rangle &
\end{tikzcd}$$

This approximation is the best approximation by open and closed elements that it is possible to define on a quasi BZ structure. To be precise, for any element $a \in X$ the following holds:

(L1) $\mathbf{i}(a)$ is an open element ($\mathbf{i}(a) \in \mathbb{O}(X)$);
(L2) $\mathbf{i}(a)$ is an open *lower* approximation of a ($\mathbf{i}(a) \leq a$);
(L3) $\mathbf{i}(a)$ is the best lower approximation of a by open elements (let $e \in \mathbb{O}(X)$ be such that $e \leq a$, then $e \leq \mathbf{i}(a)$).

By properties (L1)–(L3), it follows that the interior of an element a can be expressed as

$$\mathbf{i}(a) = \max\{x \in \mathbb{O}(X) : x \leq a\}$$

Analogously, for the closure operator,

(U1) $\mathbf{c}(a)$ is a closed element ($\mathbf{c}(a) \in \mathbb{C}(X)$).
(U2) $\mathbf{c}(a)$ is a closed *upper* approximation of a ($a \leq \mathbf{c}(a)$).
(U3) $\mathbf{c}(a)$ is the best outer approximation of a by closed elements (let $f \in \mathbb{C}(X)$ be such that $a \leq f$, then $\mathbf{c}(a) \leq f$).

By properties (U1)–(U3), it follows that the closure of an element a can be expressed as

$$\mathbf{c}(a) = \min\{y \in \mathbb{C}(X) : a \leq y\}$$

An equivalent way to define a rough approximation is to consider instead of the interior–closure pair the interior–exterior pair:

$$r_e(a) := \langle \mathbf{i}(a), \mathbf{e}(a) \rangle := \langle \mathbf{i}(a), \mathbf{c}(a)' \rangle$$

drawn in the following diagram

$$\begin{tikzcd}
& a \in \Sigma \arrow[dl, "\mathbf{i}"'] \arrow[dr, "\mathbf{e}"] \arrow[dd, "r_e"] & \\
\mathbf{i}(a) \in \mathbb{O}(\Sigma) \arrow[dr] & & \mathbf{e}(a) \in \mathbb{O}(\Sigma) \arrow[dl] \\
& \langle \mathbf{i}(a), \mathbf{e}(a) \rangle &
\end{tikzcd}$$

We remark that both $\mathbf{i}$(a) and $\mathbf{e}$(a) are open elements and that given the interior – exterior approximation $r_e(a)$, the interior – closure approximation $r(a)$ can be obtained in an obvious way, and vice versa.

Further, the interior and closure operators satisfy some typical modal properties, once interpreted the interior as a necessity operator and the closure as a possibility operator ([Che88]).

Proposition 9. *In any quasi BZ distributive lattice the following conditions hold:*

(mod–1p)

$$\mathbf{i}(1) = 1.$$

That is: if a sentence is true, then also its necessity its true (necessitation rule).

(mod–2p)

$$\mathbf{i}(a) \leq a \leq \mathbf{c}(a).$$

In other words: necessity implies actuality and actuality implies possibility (a characteristic principle of the modal system T).

(mod–3p)

$$\mathbf{i}(a) = \mathbf{i}(\mathbf{i}(a)); \quad \mathbf{c}(a) = \mathbf{c}(\mathbf{c}(a)).$$

necessity and possibility are idempotent operators (a characteristic S_4–principle).

As a consequence of these properties we can say that the interior and closure operators induced by a quasi BZ distributive lattice, present a typical behaviour of a **T** system, based on a Kleene lattice, instead of on a Boolean algebra, satisfying also the S_4–principle.

Generally, the following properties of modalities are not satisfied, as showed in Example 15.

(B) $a \leq \mathbf{i}(\mathbf{c}(a)$ (a characteristic B-principle);
(S_5) $\mathbf{i}(a) = \mathbf{c}(\mathbf{i}(a))$, $\mathbf{c}(a) = \mathbf{i}(\mathbf{c}(a))$ (a characteristic S_5-principle).

Example 15. Let us consider the quasi BZ distributive lattice given in the Hasse diagram of Figure 12.

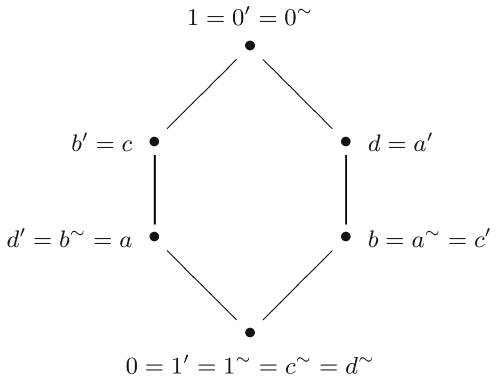


Fig. 12.

Then, we have that b is a closed element, i.e., $b = \mathbf{c}(b)$ and $\mathbf{i}(\mathbf{c}(b)) = \mathbf{i}(b) = 0$. So, $\mathbf{c}(b) \neq \mathbf{i}(\mathbf{c}(b))$ and $\mathbf{i}(\mathbf{c}(b)) < b$. Further, c is an open element, i.e., $\mathbf{i}(c) = c$ and $\mathbf{c}(\mathbf{i}(c)) = 1$. So $\mathbf{i}(c) \neq \mathbf{c}(\mathbf{i}(c))$.

Even if the interior and closure operators give the best approximation by closed and open elements, they are not the only possible ones. Another interesting possibility is given by the following two operators:

$$\nu : \Sigma \to \mathbb{C}(\Sigma) \quad \nu(a) := a'^{\sim} \tag{13a}$$

$$\mu : \Sigma \to \mathbb{O}(\Sigma) \quad \mu(a) := a^{\sim\prime} \tag{13b}$$

These mappings can be considered approximation operators because the following order chain $\nu(a) \leq a \leq \mu(a)$ holds. Moreover, they behave as *modal operators* ([Che88]), as shown in the following proposition.

Proposition 10. *In any quasi BZ distributive lattice the following conditions hold:*

(mod–1s)

$$\nu(1) = 1.$$

That is: if a sentence is true, then also its necessity its true (necessitation rule).

(mod–2s)

$$\nu(a) \leq a \leq \mu(a).$$

In other words: necessity implies actuality and actuality implies possibility (a characteristic principle of the modal system T).

(mod–3s)

$$a \leq \nu(\mu(a)).$$

Actuality implies necessity of possibility (a characteristic B-principle).

As a consequence of these properties we can say that the modal operators induced by a quasi–BZ distributive lattice, present a typical behavior of a **B** system, based on a Kleene lattice, instead of on a Boolean algebra.

Generally, ν and μ do not satisfy the S_4 and S_5 principles:

(S_4) $\nu(a) = \nu(\nu(a))$; $\mu(a) = \mu(\mu(a))$ (a characteristic S_4-principle).
(S_5) $\nu(a) = \mu(\nu(a))$, $\mu(a) = \nu(\mu(a))$ (a characteristic S_5-principle).

Example 16. Let us consider the same quasi–BZ lattice of Example 15. Then, $\nu(c) = a$, so $\nu(c) \neq \nu(\nu(c)) = 0$ and $\nu(c) \neq \mu(\nu(c)) = c$. Further, $\mu(a) = c$, so $\mu(a) \neq \mu(\mu(a)) = 1$ and $\mu(a) \neq \nu(\mu(a)) = a$.

However, necessity and possibility are not outer and inner operators. In fact, in general, they do not satisfy the idempotent properties, but only the following weaker form:

$$\nu(\nu(a)) \leq \nu(a)$$
$$\mu(a) \leq \mu(\mu(a))$$

Example 17. Let us consider the same quasi–BZ lattice of Example 15. Then, $\mu(a) = c < 1 = \mu(\mu(a))$ and $\nu(\nu(d)) = 0 < b = \nu(d)$.

The major drawback of this approximation is that the rough approximation based on modal operators is worse than the corresponding one based on the interior - closure operators, in the sense that it captures less information about the approximable elements. In fact, for any element $a \in X$, the following order chain holds:

$$\nu(a) \leq \mathbf{i}(a) \leq a \leq \mathbf{c}(a) \leq \mu(a). \tag{14}$$

3.1 Rough Approximation Spaces as Quasi-BZ Lattices

We have seen that given an element a of a quasi–BZ distributive lattice, a possible rough approximation of a is given by $r_e(a) = \langle \mathbf{i}(a), \mathbf{e}(a) \rangle$. Now, we show that it is possible to give to the collection of all rough approximations, $r_e(a)$, the structure of a quasi– BZ distributive lattice.

Let $\langle \Sigma, \wedge, \vee, ', {}^{\sim}, 0, 1 \rangle$ be a quasi–BZ lattice. For the sake of simplicity, we define

$$\forall a \in \Sigma \quad a_i := \mathbf{i}(a) \quad a_e := \mathbf{e}(a) = \mathbf{c}(a)'$$

We recall that both a_i and a_e are open elements: $a_i, a_e \in \mathbb{O}(\Sigma)$. Further, we define the collection of all rough approximations as the set $\mathbb{A} := \{\langle a_i, a_e \rangle : a \in \Sigma\}$. Then, the following proposition holds.

Proposition 11. *The structure* $\langle \mathbb{A}, \sqcap, \sqcup, ^{-}, ^{\approx}, \langle 0,1\rangle, \langle 1,0\rangle \rangle$ *where:*

$$\begin{aligned} \langle a_i, a_e\rangle \sqcap \langle b_i, b_e\rangle &:= \langle \mathbf{i}(a_i \wedge b_i), a_e \vee b_e\rangle \\ \langle a_i, a_e\rangle \sqcup \langle b_i, b_e\rangle &:= \langle a_i \vee b_i, \mathbf{i}(a_e \wedge b_e)\rangle \\ \langle a_i, a_e\rangle^{-} &:= \langle a_e, a_i\rangle \\ \langle a_i, a_e\rangle^{\approx} &:= \langle \mathbf{e}((a_e)^{\flat}), (a_e)^{\flat}\rangle \end{aligned}$$

is a quasi BZ distributive lattice.

The partial order induced by the lattice operators, $\sqcap$ and $\sqcup$, is

$$\langle a_i, a_e\rangle \sqsubseteq \langle b_i, b_e\rangle \quad \text{iff} \quad a_i \le b_i \text{ and } b_e \le a_e$$

In this environment the interior, closure and exterior operators are respectively defined as:

$$\begin{aligned} \mathbf{I}(\langle a_i, a_e\rangle) &:= \langle \mathbf{i}(a_i), \mathbf{e}(\mathbf{i}(a_i))\rangle = \langle a_i, \mathbf{e}(a_i)\rangle \\ \mathbf{C}(\langle a_i, a_e\rangle) &:= \langle \mathbf{e}(\mathbf{i}(a_e)), \mathbf{i}(a_e)\rangle = \langle \mathbf{e}(a_e), a_e\rangle \\ \mathbf{E}(\langle a_i, a_e\rangle) &:= \langle a_e, \mathbf{e}(a_e)\rangle \end{aligned}$$

The collection of open and closed sets are respectively

$$\begin{aligned} \mathbb{O}(\mathbb{A}) &= \{\langle a_i, \mathbf{e}(a_i)\rangle : a \in \Sigma\} \\ \mathbb{C}(\mathbb{A}) &= \{\langle \mathbf{e}(b_e), b_e\rangle : b \in \Sigma\} \end{aligned}$$

3.2 Preclusivity Spaces as Quasi-BZ Distributive Lattices

Now, we come back to preclusivity spaces, and analyze them in a quasi–BZ lattice perspective.

Let $\mathcal{S} = \langle X, \#\rangle$ be a preclusivity space as defined in Definition 4. As recalled in Proposition 1, the structure $\langle \mathcal{P}(X), \cap, \cup, ^{c}, ^{\#}, \emptyset, X\rangle$, is a quasi–BZ distributive lattice. Thus, the interior $\mathbf{L}_{\#}(H) = H^{c\#\#c}$ and closure $\mathbf{U}_{\#}(H) = H^{\#\#}$ operators as defined in Proposition 8 are just the preclusive lower and upper approximation as defined in Propositions 2 and 3. More interestingly, when considering the modal operators ν and μ of equations (13) it happens that they correspond to the similarity approximations given in Definition 3 by equation (2): $\nu(H) = H^{c\#} = \mathbf{L}_{\mathcal{R}}(H)$ and $\mu(H) = H^{\#c} = \mathbf{U}_{\mathcal{R}}(H)$.

As an application of the order chain (14) to the present case we have that $\forall H \subseteq X$:

$$\mathbf{L}_{\mathcal{R}}(H) \subseteq \mathbf{L}_{\#}(H) \subseteq H \subseteq \mathbf{U}_{\#}(H) \subseteq \mathbf{U}_{\mathcal{R}}(H)$$

This means that the preclusive rough approximation of a set H is always closer to H than the one given by similarity. Thus, what was just a conjecture in Example 9 has now been proved.

Now, we can apply Proposition 11 and give the structure of quasi–BZ lattices to the collection of all preclusive rough approximations. Given a set of objects H the pair interior–exterior $\langle \mathbf{i}(H), \mathbf{e}(H)\rangle$ is identified with the two sets: $\langle H_i, H_e\rangle := \langle \mathbf{L}_{\#}(H), (\mathbf{U}_{\#}(H))^c\rangle = \langle H^{c\#\#c}, H^{\#\#c}\rangle$. So, we have that the collection of all such pairs $\{\langle H_i, H_e\rangle : H \subseteq \mathcal{P}(X)\}$ is a quasi BZ distributive lattice, once defined the operators:

$$\begin{aligned}
\langle H_i, H_e\rangle \sqcap \langle K_i, K_e\rangle &:= \langle \mathbf{L}_{\#}(H_i \cap K_i), H_e \cup K_e\rangle \\
\langle H_i, H_e\rangle \sqcup \langle K_i, K_e\rangle &:= \langle H_i \cup K_i, \mathbf{L}_{\#}(H_e \cap K_e)\rangle \\
\langle H_i, H_e\rangle^{-} &:= \langle H_e, H_i\rangle \\
\langle H_i, H_e\rangle^{\approx} &:= \langle (\mathbf{U}_{\#}((H_e)^{\flat}))^c, (H_e)^{\flat}\rangle
\end{aligned}$$

4 BZ Lattices and Pawlak Rough Sets

So far we have considered concrete quasi–BZ lattices and concrete rough approximation spaces induced by a preclusivity space $\langle X, \#\rangle$, where the preclusive relation $\#$ is irreflexive and symmetric (or, equivalently, by a similarity space $\langle X, \mathcal{R}\rangle$, where the similarity relation is reflexive and symmetric). Now, in order to obtain Pawlak's rough approximation spaces ([Paw81],[Paw82], [Paw92]), it will be sufficient to assume the stronger assumption according to which in any similarity space $\langle X, \mathcal{R}\rangle$ the relation $\mathcal{R}$ is an equivalence relation (reflexive, symmetric and transitive). As a consequence, $\mathcal{R}$ will determine a partition of the universe X into a set of equivalence classes E_i: two elements x and y will be non–distinguishable ($x\mathcal{R}y$) iff they belong to one and the same equivalence class E_i. Given an element $x \in X$ we can define the *equivalence class* generated by x as the set

$$E(x) = \{y \in X : x\mathcal{R}y\}.$$

Obviously, $E(x)$ is not empty because x belongs to it and, analogously to the similarity case, it constitutes a *granule* of knowledge about x. Let us stress that two granule are either disjoint or equal between them.

In particular, in the classical Pawlak approach to rough sets the equivalence relation is based on the equality of attribute values of a *complete* information system $\mathcal{K}(X)$. Precisely, given a subset of attributes $D \subseteq Att(X)$, the equivalence relation of any two objects is defined as:

$$x \equiv y \quad \text{iff} \quad \forall a \in D \; F(x,a) = F(y,a). \tag{15}$$

Obviously, the corresponding preclusive relation is defined as:

$$x \not\equiv y \quad \text{iff} \quad \exists a \in D \; F(x,a) \neq F(y,a). \tag{16}$$

Following the general theory of Sect. 3, we have that the structure $\mathbb{P} = \langle \mathcal{P}(X), \cap, \cup, ^c, ^{\not\equiv}, \emptyset, X\rangle$ is a quasi– BZ distributive lattice. However, the transitive property of the similarity relation introduces some strong "classical" feature.

Table 15. Flats information system.

Flat	Price	Rooms	Down–Town	Furniture
f_1	high	2	yes	no
f_2	high	1	yes	no
f_3	high	2	yes	no
f_4	low	1	no	no
f_5	low	1	no	no
f_6	medium	1	yes	yes

Example 18. Let us consider the flat information system defined in Example 1 and reported below.

We set $D = Att(X)$ and $H = \{f_3, f_4\}$. Computing the similarity rough approximation
$\langle \nu(H), \mu(H) \rangle$ and the preclusivity rough approximation
$\langle \mathbf{i}(H), \mathbf{c}(H) \rangle$, we have that

$$\langle \nu(H), \mu(H) \rangle = \langle \emptyset, \{f_1, f_3, f_4, f_5\} \rangle = \langle \mathbf{i}(H), \mathbf{c}(H) \rangle$$

that is, the two approximations coincide. So, in particular, we have that

$$\nu(H) = \mathbf{i}(H)$$

that is, the necessity of H is equivalent to the interior of H, and

$$\mu(H) = \mathbf{c}(H)$$

that is, the possibility of H is equivalent to the closure of H. Further, the following properties hold: $\nu(\nu(H)) = \emptyset = \nu(H)$ and $\mu(\mu(H)) = \{f_1, f_3, f_4, f_5\} = \mu(H)$, that is, the modal operators ν and μ are idempotent.

The properties analyzed in the previous example, i.e, idempotency of necessity and possibility, equivalence of necessity (resp., possibility) and interior (resp., closure) operators, are true for any element $H \in \mathcal{P}(X)$. This is due to the fact that the system $\mathbb{P}$ satisfies the axioms of a stronger structure than quasi–BZ lattices: BZ lattices.

Definition 9. *A structure* $\langle \Sigma, \wedge, \vee, ', ^{\sim}, 0, 1 \rangle$ *is a* BZ distributive lattice *if it is a quasi–BZ distributive lattice satisfying the stronger interconnection rule:*

(s–in) $a^{\sim\sim} = a^{\sim\prime}$.

Definition 10. *A BZ distributive lattice satisfying also the* $\vee$ *de Morgan property*

(B2a) $(a \wedge b)^{\sim} = a^{\sim} \vee b^{\sim}$,

is called a de Morgan BZ (BZ^{dM}) distributive lattice.

A consequence of the new axiom (s–in) is that the necessity mapping coincides with the interior operator and the possibility mapping coincides with the closure operator.

Proposition 12. *Let $\langle \Sigma, \wedge, \vee, ', ^{\sim}, 0, 1\rangle$ be a BZ distributive lattice. Then, the following are true:*

$$\begin{aligned} \forall a \in \Sigma \quad & \nu(a) = a'^{\sim} = a'^{\sim\sim'} = \mathbf{i}(a) \\ \forall a \in \Sigma \quad & \mu(a) = a^{\sim'} = a^{\sim\sim} = \mathbf{c}(a) \end{aligned}$$

Further, we have that the modal operators, ν and μ, have the stronger behaviour of an S_5–modal like system, always based on a Kleene algebra instead of on a Boolean one. That is, beside properties (mod–1)–(mod–3), the following are satisfied:

(mod–4)

$$\nu(\nu(a)) = \nu(a)$$

$$\mu(\mu(a)) = \mu(a)$$

Necessity of necessity is equal to necessity; similarly for possibility (a characteristic S_4-principle).

(mod–5)

$$\mu(a) = \nu(\mu(a))$$

$$\nu(a) = \mu(\nu(a))$$

Possibility is equal to the necessity of possibility; whereas necessity is equal to the possibility of necessity (a characteristic S_5-principle).

Finally, another consequence of axiom (s–in) is the equivalence between the collection of open and closed sets:

$$\mathbb{O}(\Sigma) = \{a \in \Sigma : a = \mathbf{i}(a)\} = \{a \in \Sigma : a = \mathbf{c}(a)\} = \mathbb{C}(\Sigma).$$

So, we need not to distinguish between upper exact and lower exact sets, we only have exact sets, i.e., sets which are clopen. Let us indicate the collection of exact sets as Σ_e.

We remark that this is not the only way to define exact elements. We have seen that in general the order chain $\nu(a) \leq a \leq \mu(a)$ does not hold. Clearly, this is a fuzzy situation, since in a classical environment we would have no difference among necessity, actuality and possibility, i.e., we are interested in those elements for which $e = \mu(e)$ (equivalently, $e = \nu(e)$). This leads us to define the substructure of all *Modal sharp (M-sharp)* (exact, crisp) elements, denoted by $\Sigma_{e,M}$, as follows:

$$\Sigma_{e,M} := \{e \in \Sigma : \mu(e) = e\} = \{e \in \Sigma : \nu(e) = e\}.$$

However, due to the strong interconnection holding between the two orthocomplementations in a BZ structure, the collection of M–sharp elements coincides with Σ_e. Further, since in general $a \wedge a' \neq 0$ (equivalently, $a \vee a' \neq 1$) it is possible to consider as *Kleene sharp (K-sharp)* the elements which satisfy the non contradiction (or equivalently the excluded middle) law with respect to the Kleene negation:

$$\Sigma_{e,'} := \{e \in \Sigma : e \wedge e' = 0\} = \{e \in \Sigma : e \vee e' = 1\}.$$

In a BZ^{dM} distributive lattice also this set coincides with Σ_e, more generally, in a BZ distributive lattice the relation $\Sigma_e \subseteq \Sigma_{e,'}$ holds.

Summarizing, we have that in a BZ distributive lattice, we can give only one rough approximation through exact sets. The approximation operators turn out to be respectively an interior and a closure operator with the further property that they have a typical S_5 modal behaviour.

Definition 11. *Let $\langle \Sigma, \leq, ', ^{\sim}, 0, 1\rangle$ be a BZ distributive lattice. The induced rough approximation space is the structure*

$$\langle \Sigma, \Sigma_e, \nu, \mu\rangle,$$

where

- *Σ is the set of approximable elements;*
- *$\Sigma_e \subseteq \Sigma$ is the set of exact elements;*
- *$\nu : \Sigma \to \Sigma_e$ is the* inner approximation map*;*
- *$\mu : \Sigma \to \Sigma_e$ is the* outer approximation map*;*

For any element $a \in \Sigma$, its rough approximation *is defined as the pair:*

$$r(a) := \langle \nu(a), \mu(a)\rangle \quad [\textit{with} \quad \nu(a) \leq a \leq \mu(a)]$$

drawn in the following diagram:

$$\begin{tikzcd}
 & a \in \Sigma \arrow[dl, "\nu"'] \arrow[dr, "\mu"] \arrow[dd, "r"] & \\
\nu(a) \in \Sigma_e \arrow[dr] & & \mu(a) \in \Sigma_e \arrow[dl] \\
 & \langle \nu(a), \mu(a)\rangle &
\end{tikzcd}$$

Also in the case of BZ distributive lattice, $r(a)$ is the best approximation by exact elements of the element a. That is, for any element $a \in \Sigma$, properties (L1)–(L3) and (U1)–(U3) hold.

Moreover, in this context, the approximation r_e is given by the necessity–impossibility pair:

$$r_e(a) = \langle \nu(a), \mu'(a)\rangle = \langle \nu(a), a^{\sim}\rangle,$$

drawn in the following diagram:

$$\begin{tikzcd}
 & a \in \Sigma \arrow[dl, "\nu"'] \arrow[dr, "\sim"] \arrow[dd, "r"] & \\
\nu(a) \in \Sigma_e \arrow[dr] & & a^{\sim} \in \Sigma_e \arrow[dl] \\
 & \langle \nu(a), a^{\sim} \rangle &
\end{tikzcd}$$

4.1 Rough Approximation Spaces as BZ Lattices

In Sect. 3.1, we showed that it is possible to give to the collection of rough approximations induced by a quasi BZ distributive lattice the structure of a quasi–BZ distributive lattice. A dual result holds also in the case of BZ distributive lattices.

Let us consider the collection of all necessity–impossibility rough approximations. Analogously to the quasi–BZ lattice case, we define

$$\forall a \in \Sigma \quad a_i := \nu(a) \quad a_e := a^{\sim},$$

so that the collection of rough approximations is $\mathbb{A} = \{\langle \nu(a), \sim a\rangle : a \in \Sigma\} = \{\langle a_i, a_e\rangle : a \in \Sigma\}$ with a_i and a_e being exact sets: $a_i, a_e \in \Sigma_e$. Now, the following proposition holds for any BZ lattice.

Proposition 13. *The structure* $(\mathbb{A}, \sqcap, \sqcup, ^{-}, ^{\approx}, \langle 0,1\rangle, \langle 1,0\rangle)$ *where:*

$$\begin{aligned}
\langle a_i, a_e\rangle \sqcap \langle b_i, b_e\rangle &:= \langle a_i \wedge b_i, a_e \vee b_e\rangle \\
\langle a_i, a_e\rangle \sqcup \langle b_i, b_e\rangle &:= \langle a_i \vee b_i, a_e \wedge b_e\rangle \\
\langle a_i, a_e\rangle^{-} &:= \langle a_e, a_i\rangle \\
\langle a_i, a_e\rangle^{\approx} &:= \langle a_e, (a_e)'\rangle,
\end{aligned}$$

is a BZ^{dM} distributive lattice.

Also in this case the partial order induced by the lattice operators, $\sqcap$ and $\sqcup$, is

$$\langle a_i, a_e\rangle \sqsubseteq \langle b_i, b_e\rangle \quad \text{iff} \quad a_i \leq b_i \text{ and } b_e \leq a_e.$$

In this environment the interior, closure and exterior operators are respectively defined as:

$$\begin{aligned}
\mathbf{I}\langle a_i, a_e\rangle &:= \langle a_i, (a_i)'\rangle \\
\mathbf{C}\langle a_i, a_e\rangle &:= \langle (a_e)', a_e\rangle \\
\mathbf{E}\langle a_i, a_e\rangle &:= \langle a_e, (a_e)'\rangle
\end{aligned}$$

Obviously, the interior is equal to the necessity: $\mathbf{I}\langle a_i, a_e\rangle = \Box\langle a_i, a_e\rangle$, and the closure is equal to the possibility: $\mathbf{C}\langle a_i, a_e\rangle = \Diamond\langle a_i, a_e\rangle$.

As can be seen, in terms of the original BZ lattice the modal operators of necessity $\Box$ and possibility $\Diamond$ single out an exact element. That is, the necessity of a rough approximation $\langle a_i, a_e\rangle$ is equal to the rough approximation of the necessity of a:

$$\Box(r_e(a)) = \Box\,\langle a_i, a_e\rangle = \langle a_i, (a_i)'\rangle = r_e(\nu(a)) \tag{17}$$

Dually, the possibility of a rough approximation $\langle a_i, a_e\rangle$ is equal to the rough approximation of the possibility of a:

$$\Diamond(r_e(a)) = \Diamond\,\langle a_i, a_e\rangle = \langle (a_e)', a_e\rangle = r_e(\mu(a)) \tag{18}$$

The rough approximation of an element $\langle a_i, a_e\rangle$ is

$$\mathbf{R_e}\,\langle a_i, a_e\rangle = \langle \mathbf{I}\langle a_i, a_e\rangle, \mathbf{E}\langle a_i, a_e\rangle\rangle = \langle\langle a_i, (a_i)'\rangle, \langle a_e, (a_e)'\rangle\rangle$$

and according to Equations (17) and (18)

$$\mathbf{R_e}\,\langle a_i, a_e\rangle = \langle r_e(\nu(a)), r_e(\mu(a))\rangle$$

4.2 BZ Lattices Induced from Information Systems

Now, we return to Pawlak rough sets.

Proposition 14. *Let $\langle X, Att(X), val(X), F\rangle$ be a complete information system. Then, the structure $\mathbb{P} = \langle \mathcal{P}(X), \cap, \cup, ^c, ^{\not\equiv}, \emptyset, X\rangle$ where, a set of attributes $D \subseteq Att(X)$ fixed, the preclusive relation $^{\not\equiv}$,*

$$H^{\not\equiv} = \{x \in X : \forall y \in H\ \exists a \in D \quad F(x,a) \neq F(y,a)\},$$

is a BZ distributive lattice.

In general the structure $\mathbb{P}$ is not a de Morgan BZ lattice.

Example 19. Let us consider the flat information system of Table 1. We set $H = \{f_1, f_2, f_6\}$, $K = \{f_3, f_4, f_6\}$, $D = Att(X)$ and use the preclusive relation given by Equation (16). Then,

$$(H \cap K)^{\#} = \{f_6\}^{\#} = \{f_1, f_2, f_3, f_4, f_5\} \neq \{f_2, f_4, f_5\}$$

$=$

$$\{f_4, f_5\} \cup \{f_2\} = H^{\#} \cup K^{\#}.$$

In this case, we have that the necessity (resp., possibility) operator corresponds to the “classical” rough sets lower (resp., upper) approximation:

$$\nu(H) = \mathbf{L}(H) = \{x \in X : H \subseteq E(x)\}$$

$$\mu(H) = \mathbf{U}(H) = \{x \in X : H \cap E(x) \neq \emptyset\},$$

where $E(x)$ is the granule generated by the element x.

So, given a set of objects $H \subseteq X$, its rough sets approximation is given by the pair $\langle \mathbf{L}(H), \mathbf{U}(H)\rangle$. Equivalently, any rough set can be expressed as the lower–exterior pair $\langle \mathbf{L}(H), \mathbf{E}(H)\rangle = \langle \mathbf{L}(H), \mathbf{U}(H)^c\rangle$. By the previous subsection, we have that the collection of all pairs $\langle \mathbf{L}(H), \mathbf{E}(H)\rangle$ on an information system $\mathcal{K}(X) = \langle X, \mathit{Att}(X), \mathit{val}(X), F\rangle$, let us call it $\mathbb{RS}(X)$, gives rise to a BZ^{dM} distributive lattice $(\mathbb{RS}(X), \sqcap, \sqcup, {}^-, {}^{\approx}, \langle \emptyset, X\rangle, \langle X, \emptyset\rangle)$ where the operators are defined according to Proposition 13.

5 HW Algebras

In the previous section, we have seen that it is possible to give to Pawlak rough sets an algebraic representation through BZ^{dM} distributive lattices. However, this is not the only possibility. In literature, at least two other algebras are well known as a possible algebraization of rough sets. They are Heyting algebras and Wajsberg algebras ([Pag96], [Pag98], [Pol02]). In the following we will introduce Heyting Wajsberg algebras as a structure which admits as a model the collection of rough sets and as substructures all the above mentioned algebras ([CC02c]).

Definition 12. *A system* $\mathbf{A} = (A, \to_L, \to_G, 0)$ *is a* Heyting Wajsberg (*HW*) algebra *if* A *is a non empty set,* $0 \in A$, $\to_L, \to_G$ *are binary operators, such that, once defined*

1. $a \vee b := (a \to_L b) \to_L b$
2. $a \wedge b := ((a' \to_L b') \to_L b')'$
3. $a' := a \to_L 0$
4. $a^{\sim} := a \to_G 0$
5. $1 := 0'$

the following are satisfied:

(HW1) $a \to_G a = 1$
(HW2) $(a \to_G b) \wedge b = b$
(HW3) $a \to_G (b \wedge c) = (a \to_G b) \wedge (a \to_G c)$
(HW4) $a \wedge (a \to_G b) = a \wedge b$
(HW5) $(a \vee b) \to_G c = (a \to_G c) \wedge (b \to_G c)$
(HW6) $1 \to_L a = a$
(HW7) $a \to_L (b \to_L c) = (a \to_L c)' \to_L b'$
(HW8) $a^{\sim\prime} \to_L a^{\sim\sim} = 1$
(HW9) $(a \to_G b) \to_L (a \to_L b) = 1$

It turns out that the operations $\wedge$ and $\vee$ are just the meet and join operators of a distributive lattice structure whose partial order is as usual defined as:

$$a \le b \quad \text{iff} \quad a \wedge b = a \tag{19}$$

It is easy to prove that

$$a \le b \quad \text{iff} \quad a \to_L b = 1 \tag{20a}$$

$$\text{iff} \quad a \to_G b = 1 \tag{20b}$$

The primitive operator $\to_L$ behaves as a Łukasiewicz implication, as can be seen by the following proposition.

Proposition 15. *Let* $\mathbf{A}$ *be a HW algebra. Then the structure* $(A, \to_L, 1)$ *is a Wajsberg algebra according to [Waj31,Waj35], i.e., it satisfies:*

(W1) $1 \to_L a = a$
(W2) $(a \to_L b) \to_L ((b \to_L c) \to_L (a \to_L c)) = 1$
(W3) $(a \to_L b) \to_L b = (b \to_L a) \to_L a$
(W4) $(a' \to_L b') \to_L (b \to_L a) = 1$

Let us note that the structure of Wajsberg algebra induces a distributive lattice with respect to the partial order formally defined as in Equation (20a). In this lattice one can define the unary operator $a' := a \to_L 0$ which turns out to be a Kleene (or Zadeh) orthocomplementation. In other words properties (K1)–(K3) of Section 3 hold.

We remark that Wajsberg algebras are equivalent to Chang MV algebras ([Cha58]). Given a Wajsberg algerba, the MV disjunction $\oplus$ and conjunction $\odot$ operators are defined as:

$$a \oplus b := \neg a \to_L b \tag{21a}$$

$$a \odot b := \neg(a \to_L \neg b) \tag{21b}$$

Viceversa, given a MV algebra, the Łukasiewicz implication is defined as:

$$a \to b := \neg a \oplus b \tag{22}$$

On the other hand, the primitive operator $\to_G$ behaves as a Gödel implication.

Proposition 16. *Let* $\mathbf{A}$ *be a HW algebra. Then by defining the lattice operators* $\wedge$ *and* $\vee$ *as in Definition (12), the partial order relation* $\leq$ *as in Equation* (20b), *we have that* $(A, \to_G, \wedge, \vee, ')$ *is a linear symmetric Heyting algebra according to [Mon80], i.e.,* $\mathbf{A}$ *satisfies the following properties (adopting the original Monteiro enumeration):*

(A1) $a \to_G a = b \to_G b$
(A2) $(a \to_G b) \wedge b = b$
(A3) $a \to_G (b \wedge c) = (a \to_G b) \wedge (a \to_G c)$
(A4) $a \wedge (a \to_G b) = a \wedge b$
(A5) $(a \vee b) \to_G c = (a \to_G c) \wedge (b \to_G c)$
(A6) $(a')' = a$
(A7) $(a \wedge b)' = a' \vee b'$
(K) $a \wedge a' \leq b \vee b'$
(L) $(a \to_G b) \vee (b \to_G a) = 1$

In particular, the unary operation $a^{\sim} = a \to_G 0$ is a *Brouwer* orthocomplementation. In other words properties (B1)–(B3) of Section 3 hold .

Finally, from the above propositions we have that any HW algebra induces a lattice structure endowed with a Kleene and a Brouwer negation, then it can be easily proved that HW algebras have as substructures distributive BZ^{dM} lattices.

Proposition 17. *Let* **A** *be a HW algebra. Then, by defining* $\wedge$ *and* $\vee$ *as in (12), we have that* $(A, \wedge, \vee, ', ^{\sim}, 0, 1)$ *is a distributive BZ*^{dM} *lattice.*

As a consequence of the latter proposition, we can define in any HW algebra a possibility (resp., necessity) operator and, as in BZ lattices, it behaves as a closure (resp., interior) operator. Further, these operators have an S_5 modal behaviour. So, in any HW algebra we can define a rough approximation space as in Definition 11. We recall that, given an element $a \in A$ its rough approximation is given by the necessity–possibility pair:

$$r(a) := \langle \nu(a), \mu(a) \rangle = \langle a'^{\sim}, a^{\sim\prime} \rangle \quad [\text{with} \quad \nu(a) \leq a \leq \mu(a)],$$

or equivalently, by the necessity–impossibility pair:

$$r_e(a) := \langle \nu(a), \mu'(a) \rangle = \langle a'^{\sim}, a^{\sim} \rangle.$$

In the following we will see that it is possible to give the structure of a HW algebra to the collection of all rough approximations that can be obtained on a given BZ lattice, and consequently on a given HW algebra.

5.1 Rough Approximation Spaces as HW Algebras

In Section 4.1, we have proved that it is possible to give to the collection of all rough approximations $\mathbb{A} := \{\langle \nu(a), a^{\sim} \rangle : a \in \Sigma\}$ on a BZ lattice $\langle \Sigma, \wedge, \vee, ', ^{\sim}, 0, 1 \rangle$ the structure of a BZ^{dM} lattice. We now prove that a stronger result holds that $\mathbb{A}$ has the structure of an HW algebra.

Let us set, for the sake of simplicity $a_i = \nu(a)$ and $a_e = \mu(a)' = a^{\sim}$, and consider the collection of all necessity–impossibility pairs on a BZ distributive lattice $\mathbb{A} := \{\langle a_i, a_e : a \in \Sigma\}$. Then the following result holds:

Proposition 18. *The structure* $(\mathbb{A}, \Rightarrow_L, \Rightarrow_G, \mathbf{0})$*, where* $\Rightarrow_L, \Rightarrow_G$ *and* $\mathbf{0}$ *are defined as*

$$\langle a_i, a_e \rangle \Rightarrow_L \langle b_i, b_e \rangle := \langle ((a_i)' \wedge (b_e)') \vee a_e \vee b_i, a_i \wedge b_e \rangle \tag{23a}$$

$$\langle a_i, a_e \rangle \Rightarrow_G \langle b_i, b_e \rangle := \langle ((a_i)' \wedge (b_e)') \vee a_e \vee b_i, (a_e)' \wedge b_e \rangle \tag{23b}$$

$$\mathbf{0} := \langle 0, 1 \rangle \tag{23c}$$

is a HW algebra.

According to Definition 12, it is possible to define the lattice operators on $\mathbb{A}$ as:

$$\langle a_i, a_e \rangle \sqcup \langle b_i, b_e \rangle := (\langle a_i, a_e \rangle \Rightarrow_L \langle b_i, b_e \rangle) \Rightarrow_L \langle b_i, b_e \rangle = \langle a_i \vee b_i, a_e \wedge b_e \rangle$$

$$\langle a_i, a_e \rangle \sqcap \langle b_i, b_e \rangle := ((\langle a_i, a_e \rangle^{-} \Rightarrow_L \langle b_i, b_e \rangle^{-}) \Rightarrow_L \langle b_i, b_e \rangle^{-})^{-} = \langle a_i \wedge b_i, a_e \vee b_e \rangle,$$

whose induced partial order is, as usual:

$$\langle a_i, a_e \rangle \sqsubseteq \langle b_i, b_e \rangle \quad \text{iff} \quad a_i \leq b_i \text{ and } b_e \leq a_e.$$

According to Equations (21), the MV disjunction and conjunction operators are defined as:

$$\langle a_i, a_e\rangle \oplus \langle b_i, b_e\rangle := \langle (a_e^c \wedge b_e^c) \vee (a_i \vee b_i), a_e \wedge b_e\rangle$$
$$\langle a_i, a_e\rangle \odot \langle b_i, b_e\rangle := \langle a_i \wedge b_i, (a_i^c \wedge b_i^c) \vee (a_e \vee b_e)\rangle$$

The Kleene and Brouwer negations, the modal operators and the rough approximation are defined as in Subsection 4.1.

Of course, Theorem 18 holds for any BZ lattice. So, in particular, we can consider the structure $\mathbb{P} = \langle \mathcal{P}(X), \cap, \cup, ^c, ^{\not\equiv}, \emptyset, X\rangle$ which is a BZ lattice thanks to Proposition 14. As a consequence, we have that the collection of Pawlak rough sets $\mathbb{RS}(X)$ on an information system $\mathcal{K}(X) = \langle X, \mathit{Att}(X), \mathit{val}(X), F\rangle$ has an HW algebraic structure $\langle \mathbb{RS}(X), \Rightarrow_L, \Rightarrow_G, \mathbf{0}\rangle$ once operators $\Rightarrow_L$ and $\Rightarrow_G$, and the constant $\mathbf{0}$ are properly defined according to Proposition 18.

Example 20. Let us consider the flats information system introduced in Table 1. Once defined the sets $H = \{f_1, f_4, f_5\}$, $K = \{f_2, f_4, f_6\}$ and considering all the attributes to define the equivalence relation $\mathcal{R}_D$, i.e. $D = \mathit{Att}(X)$, we have that the rough sets induced by H and K are respectively: $r_e(H) = \langle \{f_4, f_5\}, \{f_2, f_6\}\rangle$ and $r_e(K) = \langle \{f_2, f_6\}, \{f_1, f_3\}\rangle$. So the HW implications applied to $r_e(H)$ and $r_e(K)$ become:

$$\langle H_i, H_e\rangle \rightarrow_L \langle K_i, K_e\rangle = \langle \{f_2, f_6\}, \emptyset\rangle$$
$$\langle H_i, H_e\rangle \rightarrow_G \langle K_i, K_e\rangle = \langle \{f_2, f_6\}, \{f_1, f_3\}\rangle$$

Needless to stress $\emptyset \subseteq \{f_1, f_3\}$ so it is verified that

$$(\langle H_i, H_e\rangle \rightarrow_G \langle K_i, K_e\rangle) \sqsubseteq (\langle H_i, H_e\rangle \rightarrow_L \langle K_i, K_e\rangle)$$

The meet and join lattice operators are:

$$\langle H_i, H_e\rangle \sqcup \langle K_i, K_e\rangle = \langle \{f_2, f_4, f_5, f_6\}, \emptyset\rangle$$
$$\langle H_i, H_e\rangle \sqcap \langle K_i, K_e\rangle = \langle \emptyset, \{f_1, f_2, f_3, f_6\}\rangle$$

As can be seen, $r_e(H)$ and $r_e(K)$ are incomparable, i.e. neither $r_e(H) \sqsubseteq r_e(K)$ nor $r_e(K) \sqsubseteq r_e(H)$.
The MV disjunction and conjunction are respectively:

$$\langle H_i, H_e\rangle \oplus \langle K_i, K_e\rangle = \langle \{f_1, f_3, f_4, f_5\}, \emptyset\rangle$$
$$\langle H_i, H_e\rangle \odot \langle K_i, K_e\rangle = \langle \emptyset, \{f_1, f_2, f_3, f_6\}\rangle$$

Finally, the necessity and possibility of $r_e(H)$ are respectively:

$$\Box\langle H_i, H_e\rangle = \langle \{f_4, f_5\}, \{f_1, f_2, f_3, f_6\}\rangle$$
$$\Diamond\langle H_i, H_e\rangle = \langle \{f_1, f_3, f_4, f_5\}, \{f_2, f_6\}\rangle$$

6 Fuzzy Sets

In the previous sections we introduced some different algebraic approaches to the theory of rough sets. In particular, we have seen that it is possible to give a rich algebraic structure to the collection of rough sets on an information system. Now, we see how the same approach applies also to fuzzy sets by showing that the collection of fuzzy sets, once suitable operators are defined, has the structure of an HW algebra. This result leads us to the possibility of having a complete algebra which is able to fully characterize both rough and fuzzy sets.

Let us consider a set of objects X, called the Universe. A *fuzzy set* or *generalized characteristic functional* on X is defined as usual as a $[0, 1]$–valued function on X:

$$f : X \mapsto [0, 1]$$

We will indicate the collection of all fuzzy sets on X as $\mathcal{F}(X) = [0, 1]^X$. This set contains two special elements: the identically 0 fuzzy set (also characteristic function of the empty set: $\underline{0} = \chi_\emptyset$)

$$\forall x \in X \quad \underline{0}(x) := 0$$

and the identically one fuzzy set (also characteristic function of the universe : $\underline{1} = \chi_X$)

$$\forall x \in X \quad \underline{1}(x) := 1$$

Obviously, this two elements will play the role of minimum and maximum element of the lattice structure we are going to define.

Proposition 19. *The structure $\langle [0, 1]^X, \rightarrow_L, \rightarrow_G, \underline{0} \rangle$ is a HW algebra once defined the implication operators as follows*

$$(f_1 \rightarrow_L f_2)(x) := \min\{1, 1 - f_1(x) + f_2(x)\}$$

$$(f_1 \rightarrow_G f_2)(x) := \begin{cases} 1 & f_1(x) \leq f_2(x) \\ f_2(x) & \textit{otherwise} \end{cases}$$

Example 21. Let us define the fuzzy set of *tall* and *very tall* people. So, we have that x is a real positive number, representing the height in centimeters. The tall fuzzy set is defined as:

$$t(x) := \begin{cases} 1 & x \geq 1.80 \\ \frac{x-1.50}{0.30} & 1.50 < x < 1.80 \\ 0 & x \leq 1.50 \end{cases}$$

and the very tall people fuzzy set is:

$$vt(x) := \begin{cases} 1 & x \geq 1.90 \\ \frac{x-1.60}{0.30} & 1.60 < x < 1.90 \\ 0 & x \leq 1.60 \end{cases}$$

If we compute the degree in which the fuzzy set very tall implies the fuzzy set tall, we correctly obtain with both implications the value 1:

$$vt \to_L t = \underline{1}$$
$$vt \to_G t = \underline{1}$$

that is, if one is very tall is surely tall. Of course, the viceversa is not true, and computing the implication of tall to very tall, we obtain:

$$(t \to_L vt)(x) := \begin{cases} 1 & x \leq 1.50 \quad \text{or} \quad x \geq 1.90 \\ \frac{1.80-x}{0.30} & 1.50 < x \leq 1.60 \\ \frac{2}{3} & 1.60 \leq x < 1.80 \\ \frac{x-1.60}{0.30} & 1.80 \leq x < 1.90 \end{cases}$$

in the case of $\to_L$ implication and

$$(t \to_G vt)(x) := \begin{cases} 1 & x \leq 1.50 \quad \text{or} \quad x \geq 1.90 \\ 0 & 1.50 < x \leq 1.60 \\ \frac{x-1.60}{0.30} & 1.60 \leq x < 1.90 \end{cases}$$

in the case of $\to_G$ operator. We remark that it holds an order relation among the two fuzzy sets $(t \to_G vt)$ and $(t \to_L vt)$: the first one is always less than or equal to the second one, as can be easily seen in Figure 13.

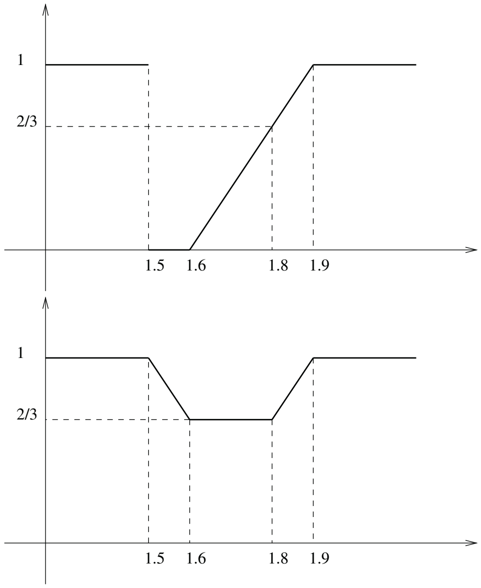


Fig. 13. On the top it is drawn the set $(t \to_G vt)$ and on the bottom the set $(t \to_L vt)$.

The lattice operators on $\mathcal{F}$ are defined as:

$$(f_1 \vee f_2)(x) := \max\{f_1(x), f_2(x)\}$$
$$(f_1 \wedge f_2)(x) := \min\{f_1(x), f_2(x)\}$$

The partial order relation induced on $\mathcal{F}$ by the above lattice operators is the usual pointwise ordering on fuzzy sets:

$$\forall f_1, f_2 \in [0,1]^X, \quad f_1 \leq f_2 \text{ iff } \forall x \in X, f_1(x) \leq f_2(x)$$

The MV operators of conjunction and disjunction are respectively defined as:

$$(f_1 \oplus f_2)(x) := \min\{1, f_1(x) + f_2(x)\}$$
$$(f_1 \odot f_2)(x) := \max\{0, f_1(x) + f_2(x) - 1\}$$

The Zadeh, Brouwer and anti–Brouwer negations are respectively defined as:

$$f'(x) := 1 - f(x)$$
$$f^{\sim}(x) := \begin{cases} 1, & \text{if } \ f(x) = 0 \\ 0, & \text{otherwise} \end{cases}$$
$$(\flat f)(x) := \begin{cases} 1 & f(x) \neq 1 \\ 0 & \text{otherwise} \end{cases}$$

For any fuzzy set $f \in [0,1]^X$ one can single out some peculiar subsets of the universe X:

- The *certainly–yes* domain of f:

$$\Delta_1(f) := \{x \in X : f(x) = 1\}$$

- The *certainly–no* domain of f:

$$\Delta_0(f) := \{x \in X : f(x) = 0\}$$

The necessity of a fuzzy set f is the characteristic functional of the certainly–yes domain of f:

$$\nu(f) = \chi_{\Delta_1(f)} = \begin{cases} 1 & \text{if } f(x) = 1 \\ 0 & \text{otherwise} \end{cases}$$

We can assume that two fuzzy sets *define the same property* iff they have the same certainly–yes domain. In this way, any property of fuzzy sets is associated with the certainly–yes domain $\Delta_1(f)$ of any of its elements. Such a property is interpreted as follows: "*belonging with certainty to the subset $\Delta_1(f)$ of X*". Consequently any property is exactly represented by the characteristic functional $\chi_{\Delta_1(f)}$.

The possibility of a fuzzy set f is

$$\mu(f) = \begin{cases} 1 & \text{if } f(x) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Hence the impossibility operator is given by the characteristic functional of the certainly–no domain of f:

$$\mu'(f) = \chi_{\Delta_0(f)}.$$

We can assume that two fuzzy sets *define the same noperty* iff they have the same certainly–no domain. Any noperty can be associated with the unique subset $\Delta_0(f)$ of X that represents the noperty in question: "*not belonging with certainty to the subset $\Delta_0(f)$ of X*". As a consequence, our noperty is exactly represented by the characteristic functional $\chi_{\Delta_0(f)}$

References

[Cat97] G. Cattaneo. Generalized rough sets (preclusivity fuzzy-intuitionistic BZ lattices). *Studia Logica* **58**, 1997, pp. 47–77.

[Cat98] ______. Abstract approximation spaces for rough theories, in: *Rough Sets in Knowledge Discovery 1: Methodology and Applications* (L. Polkowski, A.Skowron (eds.)). Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 1998, pp. 59–98.

[CC02a] G. Cattaneo, D. Ciucci. A quantative analysis of preclusivity vs. similarity based rough approximations. *Lecture Notes in Artificial Intelligence* **2475**, Springer, 2002, pp. 69–76.

[CC02b] ______. BZW algebras for an abstract approach to roughness and fuzziness, in: *Proceedings IPMU 2002*, Annecy, France, 2002, pp. 1103–1110.

[CC02c] ______. Heyting Wajsberg algebras as an abstract environment linking fuzzy and rough sets. *Lecture Notes in Artificial Intelligence* **2475**, Springer, 2002, pp. 77–84.

[CCG98] G. Cattaneo, M. L. Dalla Chiara, and R. Giuntini. Some algebraic structures for many–valued logics. *Tatra Mountains Mathematical Publications* **15**, 1998, pp. 173–196 (Special Issue: Quantum Structures II, Dedicated to Gudrun Kalmbach).

[CGP99] G. Cattaneo, R. Giuntini, and R. Pilla. BZMV^{dM} and Stonian MV algebras (applications to fuzzy sets and rough approximations). *Fuzzy Sets and Systems* **108**, 1999, pp. 201–222.

[Cha58] C. C. Chang. Algebraic analysis of many valued logics. *Trans. Amer. Math. Soc.* **88**, 1958, pp. 467–490.

[Che88] B. F. Chellas. *Modal Logic, An Introduction.* Cambridge Univ. Press, Cambridge, 1988.

[DO01] I. Düntsch, E. Orlowska. Beyond modalities: Sufficiency and mixed algebras, in: *Relational Methods for Computer Science Applications* (E. Orlowska, A. Szalas (eds.)). Physica, Heidelberg, 2001, pp. 277–299.

[GMS98] S. Greco, B. Matarazzo, and R. Slowinski. Fuzzy similarity relation as a basis for rough approximations, in: *Rough Sets and Current Trends in Computing, First International Conference, RSCTC'98*, Warsaw,(L. Polkowski, A. Skowron (eds.)). *Lecture Notes in Artificial Intelligence* **1424**, Springer, 1998, pp. 283–289.

[Kel55] J.L. Kelley. *General Topology.* Springer, New York, 1955.

[Kry98] M. Kryszkiewicz. Rough set approach to incomplete information systems. *Information Sciences* **112**, 1998, pp. 39–49.

[Mon80] A. A. Monteiro. Sur les algèbres de Heyting symétriques. *Portugaliae Mathematica* **39**, 1980, pp. 1–237.

[NSS98] S. Hoa Nguyen, A. Skowron, and P. Synak. Discovery of data pattern with applications to decomposition and classification problems, in: *Rough Sets in Knowledge Discovery 2* (L. Polkowski, A. Skowron (eds.)). Physica, Heidelberg, 1998, pp. 55–97.

[Orl98a] E. Orlowska (ed.). *Incomplete information: Rough Set Analysis.* Physica, Heidelberg, 1998.

[Orl98b] E. Orłowska. Introduction: What you always wanted to know about rough sets, in: *Incomplete Information: Rough Set Analysis* [Orl98a], pp. 1–20.

[Pag96] P. Pagliani. Rough sets and Nelson algebras. *Fundamenta Informaticae* **27**, 1996, pp. 205–219.

[Pag98] ______. Rough set theory and logic–algebraic structures, in: [Orl98a], pp. 109–190.

[Paw81] Z. Pawlak. Information systems - theoretical foundations. *Information Systems* **6**, 1981, pp. 205–218.

[Paw82] ______. Rough sets. *International Journal of Information and Computer Sciences* **11**, 1982, pp. 341–356.

[Paw85] ______. Rough sets and fuzzy sets. *Fuzzy Sets and Systems* **17**, 1985, pp. 99–102.

[Paw92] ______. Rough sets: A new approach to vagueness, in: *Fuzzy Logic for the Management of Uncertainty* (L. A. Zadeh, J. Kacprzyk (eds.)). Wiley, New York, 1992, pp. 105–118.

[Paw98] ______. Rough set elements, in: *Rough Sets in Knowledge Discovery* (L. Polkowski, A. Skowron (eds.)). Studies in Fuzziness and Soft Computing Series, Physica, Heidelberg, 1998, pp. 10–30.

[Pol02] L. Polkowski. *Rough Sets. Mathematical Foundations.* Physica, Heidelberg, 2002.

[SS96] A. Skowron, J. Stepaniuk. Tolerance approximation spaces. *Fundamenta Informaticae* **27**, 1996, pp. 245–253.

[ST99] J. Stefanowski, A. Tsoukiàs. On the extension of rough sets under incomplete information. *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Proceedings of the 7th International Workshop, RSFDGrC '99*, Yamaguchi, Japan (S. Ohsuga, Ning Zhong, and A. Skowron (eds.)). *Lecture Notes in Artificial Intelligence* **1711**, Springer, 1999, pp. 73–81.

[ST01] J. Stefanowski, A. Tsoukiàs. Valued tolerance and decision rules,'. *Lecture Notes in Artificial Intelligence* **2005**, Springer, 2001, pp. 212–219.

[Ste98] J. Stepaniuk. Approximation spaces in extensions of rough sets theory, in: *Proceedings of Rough Sets and Current Trends in Computing, First International Conference, RSCTC'98* (Warsaw, Poland) (L. Polkowski, A. Skowron (eds.)). *Lecture Notes in Artificial Intelligence* **1424**, Springer, 1998, pp. 290–297.

[Sur77] S. Surma (ed.). *Logical Works of Jan Łukasiewicz.* Polish Academy of Sciences, Wroclaw, 1977.

[SV95] R. Słowinski, D. Vanderpooten. Similarity relation as a basis for rough approximations. Tech. Report 53, Institute of Computer Science, University of Technology, Warsaw, 1995.

[Vak91] D. Vakarelov. A modal logic for similarity relations in Pawlak knowledge representation systems. *Fundamenta Informaticae* **15**, 1991, pp. 61–79.

[Waj31] M. Wajsberg. Aksjomatyzacja trówartościowego rachunku zdań [Axiomatization of three–valued propositional calculus]. *Comptes Rendus des Séances de la Societé des Sciences et des Lettres de Varsovie* **24**, 1931, pp. 126–148 (eng. transl. [Sur77]).

[Waj35] M. Wajsberg. Beiträge zum Metaaussagenkalkül I. *Monatshefte fur Mathematik un Physik* **42**, 1935, pp. 221–242 (eng. transl. [Sur77]).

Rough Mereology as a Link Between Rough and Fuzzy Set Theories. A Survey

Lech Polkowski[1,2]

[1] Polish–Japanese Institute of Information Technology
Koszykowa 86, 02 008 Warsaw, Poland
[2] Dept. Math.Computer Science, Univ. Warmia and Mazury
Zolnierska 14 A, 10 561 Olsztyn, Poland
Lech.Polkowski@pjwstk.edu.pl

Abstract. In this paper, we discuss rough inclusions defined in Rough Mereology – a paradigm for approximate reasoning introduced by Polkowski and Skowron – as a basis for common models for rough as well as fuzzy set theories. We justify the point of view that tolerance (or, similarity) is the motif common to both theories. To this end, we demonstrate in Sect. 6 that rough inclusions (which represent a hierarchy of tolerance relations) induce rough set theoretic approximations as well as partitions and equivalence relations in the sense of fuzzy set theory. Before that, we include an account of mereological theory in Sect. 3. We also discuss granulation mechanisms based on rough inclusions with an outline of applications to rough–fuzzy–neurocomputing and computing with words in Sects. 4 and 5.

Keywords: rough set theory, fuzzy set theory, rough mereology, rough inclusions, granular calculus, rough-fuzzy-neurocomputing, computing with words

1 Introduction

We begin with a concise review of both theories.

1.1 Rough Sets: Basic Ideas

Rough Set Theory begins with the idea, [16], [17], of an *approximation space*, understood as a universe U together with a family $\mathcal{R}$ of equivalence relations on U (knowledge base). Given a sub–family $\mathcal{S} \subseteq \mathcal{R}$, the equivalence relation $S = \bigcap \mathcal{S}$ induces a partition P_S of U into equivalence classes $[x]_S$ of the relation S.

In terms of P_S, *concept approximation* is possible; a *concept relative to* U (or, shortly, a U*–concept*) is a subset $X \subseteq U$. There are two cases.

A U–concept X is S*–exact* in case,

$$X = \bigcup \{[x]_S : [x]_S \subseteq X\}, \tag{1}$$

holds. We will speak in this case also of an S*–exact set*.

Otherwise, X is said to be an S*–rough* U-concept (or, set).

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 253–277, 2004.

In the second case, the idea of an *approximation* comes useful [16]. Two S–exact sets, approximating X from below and from above, are the *lower S–approximation*,

$$\underline{S}X = \bigcup\{[x]_S : [x]_S \subseteq X\}, \tag{2}$$

and the *upper S–approximation*,

$$\overline{S}X = \bigcup\{[x]_S : [x]_S \cap X \neq \emptyset\}. \tag{3}$$

Then, clearly,

1. $\underline{S}X \subseteq X \subseteq \overline{S}X$.
2. $\underline{S}X$ (respectively, $\overline{S}X$) is the largest (respectively, the smallest) S–exact set contained in (respectively, containing) X.

Sets (U–concepts) with identical approximations may be identified; consider an equivalence relation $\approx_S$ defined as follows [18]:

$$X \approx_S Y \Leftrightarrow \underline{S}X = \underline{S}Y \wedge \overline{S}X = \overline{S}Y. \tag{4}$$

This is clearly an equivalence relation; let $Concepts_S$ denote the set of these classes. Then for $x, y \in Concepts_S$, we have

$$x = y \Leftrightarrow \forall u \in U.\phi_u(x) = \phi_u(y) \wedge \psi_u(x) = \psi_u(y), \tag{5}$$

where for $x = [X]_{\approx_S}$, $\phi_u(x) = 1$ in case $[u]_S \subseteq X$, otherwise $\phi_u(x) = 0$; similarly, $\psi_u(x) = 1$ when $[u]_S \cap X \neq \emptyset$, otherwise $\psi_u(x) = 0$.

The formula (5) witnesses *Leibnizian indiscernibility* in $Concept_S$: entities are distinct if and only if they are discerned by at least one of available functionals (in our case, ϕ_u, ψ_u). The idea of **indiscernibility** is one of the most fundamental in Rough Set Theory [16].

Other fundamental notions are derived from the observation on complexity of the generation of S: one may ask whether there is some $\mathcal{T} \subset \mathcal{S}$ such that

$$S = \bigcap \mathcal{T}. \tag{6}$$

In case the answer is positive, one may search for a minimal with respect to inclusion subset $\mathcal{T} \subseteq \mathcal{S}$ satisfying (6). Such a subset is said to be an S–**reduct**. Let us observe that $\mathcal{T} \subseteq \mathcal{S}$ is an S–reduct if and only if for each $R \in \mathcal{T}$ we have,

$$\bigcap(\mathcal{T} \setminus \{R\}) \neq \bigcap \mathcal{T}. \tag{7}$$

In this case we say that $\mathcal{U}$ is **independent**; this notion falls under a general independence scheme [14].

These ideas come fore most visibly in case of knowledge representation in the form of *information systems* [16]. An *information system* is a universe U along with a set A of *attributes* each element a of which is a mapping $a : U \to V_a$ from

U into an *a–value set* V_a. Clearly, each attribute $a \in A$ does induce a relation of *a–indiscernibility* $IND(a)$ defined as follows,

$$xIND(a)y \Leftrightarrow a(x) = a(y). \tag{8}$$

The family $\{Ind(a) : a \in A\}$ is a knowledge base and for each $B \subseteq A$, the relation $IND(B) = \bigcap\{IND(a) : a \in B\}$ is defined, inducing the lower and the upper approximations $\underline{B}X, \overline{B}X$. Notions of a B–reduct, and B–independence are defined as in general case [16].

From now on, we will work with the reduced universe $U/IND(A)$ without mentioning this fact explicitly; thus, the identity $=_U$ on objects – classes representatives – will denote the equality of indiscernibility classes.

1.2 Fuzzy Sets: Basic Notions

A starting point for Fuzzy Set Theory is that of a *fuzzy set* [26]. Fuzzy sets come as a generalization of the usual mathematical idea of a set: given a universe U, a set X in U may be expressed by means of its *characteristic function* χ_X: $\chi_X(u) = 1$ in case $u \in X$, $\chi_X(u) = 0$, otherwise.

A fuzzy set X is defined by allowing χ_X to take values in the interval $[0, 1]$. Thus, $\chi_X(u) \in [0, 1]$ is a measure of *degree to which u is in X*. Once fuzzy sets are defined, one may define a *fuzzy algebra of sets* by defining operators responsible for the union, the intersection, and the complement in the realm of fuzzy sets. Usually, those are defined by selecting a *t–norm*, a *t–co–norm*, and a *negation* functions where a t–norm $T(x, y)$ is a function allowing the representation (see [13], cf. [21] Ch. 14),

$$T(x, y) = g(f(x) + f(y)), \tag{9}$$

where the function $f : [0, 1] \to [0, +\infty)$ in (9) is continuous decreasing on $[0, 1]$ and g is the pseudo–inverse to f (i.e. $g(u) = 0$ in case $u \in [0, f(1)]$, $g(u) = f^{-1}(u)$ in case $u \in [f(1), f(0)]$, and $g(u) = 1$ in case $u \in [f(0), +\infty)$).

A t–conorm C is induced by a t–norm T via the formula $C(x, y) = 1 - T(1 - x, 1 - y)$. A negation $n : [0, 1] \to [0, 1]$ is a continuous decreasing function such that $n(n(x)) = x$. An important example of a t–norm is the *Łukasiewicz product*,

$$\otimes(x, y) = max\{0, x + y - 1\}; \tag{10}$$

we recall also the *Menger product*,

$$Prod(x, y) = x \cdot y. \tag{11}$$

1.3 Rough Membership Functions

In a rough universe represented by an information system (U, A), it is possible to introduce, as shown by Pawlak and Skowron [19], a parameterized family of functions measuring for a given object $u \in U$ and a set $X \subseteq U$ degree to which

u belongs in X, a parameter being a subset $B \subseteq A$ of attributes. In this case, the *rough membership function* $m_{B,X}$ on U is defined as,

$$m_{B,X}(u) = \frac{|X \cap [u]_B|}{|[u]_B|}. \tag{12}$$

It may be noticed that [19],

1. In case X is a B–exact set, $m_{B,X}$ is the characteristic function of X, i.e., X is perceived as a crisp set;
2. $m_{B,X}$ is a piece–wise constant function, constant on classes of $IND(B)$.

From $m_{B,X}$, rough set approximations are reconstructed as follows,

$$\underline{B}X = \{u \in U : m_{B,X}(u) = 1\}, \tag{13}$$

$$\overline{B}X = \{u \in U : m_{B,X}(u) > 0. \tag{14}$$

Rough membership functions are prototypes of *rough inclusions* defined in the sequel which measure degrees to which one object (concept) is contained in another, and containment is expressed in terms of relations of being a part.

Let us observe a contextual character of rough membership functions: they are defined relative to sets of attributes, hence, they represent information content of those sets. A similar character will be featured in rough inclusions defined in the framework of information systems.

2 Rough Mereology

We introduce a reasoning mechanism based on Rough Mereology [22], whose distinct facets correspond to rough respectively fuzzy approaches to reasoning allowing on one hand the introduction of rough set–theoretic approximations and on the other hand inducing fuzzy set–theoretic notions of a partition as well as of an equivalence relation. We will present the main ideas of this approach.

2.1 Mereology

Mereological theories, contrary to naive or formalized set theories that are based on the notion of being an element of a set, are based on the idea that relations among object should be based on their containment properties (e.g., we cannot say "the circle is an element of the closed disk it is boundary of" but we can say "the circle is the part of the closed disk"). Mereology theory proposed by Stanisław Leśniewski proposes the notion of a *part* as the primitive one, whereas the mereological theory outlined by Alfred North Whitehead [24] and developed by Leonard, Goodman, and Clarke, see [4], among others, begins with the notion of *connection*.

We work here in the formalism proposed by Leśniewski, and thus the basic notion is that of a *part* relation on a universe U with identity $=$, [11]; in symbols:

$x\pi y$, that reads x *is a part of* y. A part relation π satisfies by definition the following conditions,

$$x\pi y \wedge y\pi z \Rightarrow x\pi z, \tag{15}$$

$$\neg(x\pi x). \tag{16}$$

Thus, the part relation is transitive and non–reflexive (i.e., it is a pre–order on U), so it can be interpreted as expressing the idea of a *proper part*.

The idea of an improper part, i.e., possibly of the whole object, is expressed by the notion of an *element*, that may not be confused with the notion of an element in naive set theory (sometimes, the term “ingredient” is used in place of that of “element”, such was original usage of Leśniewski).

The notion of an *element* relation, el_π, induced by the relation π is the following,

$$x\ el_\pi\ y \Leftrightarrow x\pi y \vee x = y. \tag{17}$$

Thus, $x\ el_\pi\ y \wedge y\ el_\pi\ x$ is equivalent to $x = y$. By (15), (16), the relation el_π is an ordering on the universe U. Moreover,

$$xel_\pi x \tag{18}$$

for each x in U, a striking difference with the usage of the notion of an element in naive set theory.

The fundamental feature of mereology of Leśniewski is that it is concerned with collective classes, i.e., distributive classes (names, concepts) are made into objects (collective classes).

In order to make a non–empty concept $M \subseteq U$ into a collective object, the *class* operator Cls is used [11].

The definition of the class of M, in symbols $Cls(M)$, is as follows,

$$x \in M \Rightarrow x\ el_\pi\ Cls(M), \tag{19}$$

$$x\ el_\pi\ Cls(M) \Rightarrow \exists y, z. y\ el_\pi\ x \wedge y\ el_\pi\ z \wedge z \in M. \tag{20}$$

Condition (19) includes all members of M into $Cls(M)$ as elements; (20) requires each element of $Cls(M)$ to have an element in common with a member of M (compare this with the definition of the union of a family of sets to see the analogy that becomes clear, when the relation of being a (mereological) element is interpreted as that of being a subset of a set, and M is a family of subsets in a given universe).

One requires also $Cls(M)$ to be unique, [11]. From this demand it follows that, [11],

$$[\forall y.(y\ el_\pi\ x \Rightarrow \exists w.w\ el_\pi\ y \wedge w\ el_\pi\ z)] \Rightarrow x\ el_\pi\ z. \tag{21}$$

The rule (21) is useful for recognizing that $x\ el_\pi\ z$, and it will be used in our arguments.

2.2 Rough Mereology

Given a mereological universe (U, π), with the induced relation el_π of being an element, we introduce, cf. [22], on $U \times U \times [0,1]$ a relation $\mu(x, y, r)$, read *x is a part of y to degree at least r*. We will write rather more suggestively $x\mu_r y$, calling μ a *rough inclusion* (it is the generic name introduced in [22]).

We introduce at this moment an element of ontology (we do not enter here any discussion on what is ontology; here, by ontology we mean simply a system of names we use for naming various types of objects): we will regard U as a name for atomic objects, and the formula $x\varepsilon U$ (read: *x is U*) will mean that an object x is an atomic object.

In case of an information system (U, A), atomic objects will be objects $x \in U$. The symbol G_U will name individual objects constructed as classes of atomic objects, which will be denoted with the formula $g\varepsilon G_U$, i.e., $g\varepsilon G_U$ if and only if there exists a non–empty property Φ with $g = Cls(\Phi|U)$, $\Phi|U$ meaning Φ restricted to atomic objects, $\Phi|U$ a non–empty property of atomic objects.

We require μ to satisfy the following conditions, see [22],

$$x\mu_1 x, \tag{22}$$

$$x\mu_1 y \Leftrightarrow x \ el_\pi \ y, \tag{23}$$

$$x\mu_1 y \Rightarrow \forall z, r.(z\mu_r x \Rightarrow z\mu_r y), \tag{24}$$

$$x\mu_r y \wedge s < r \Rightarrow x\mu_s y. \tag{25}$$

Informally, (23) ties the rough inclusion to the mereological underlying universe , (24) does express monotonicity (a bigger entity cuts a bigger part of everything), (25) says that a part to degree r is a part to any lesser degree.

We may observe that (22) is a consequence to (23) due to (18).

2.3 More Ontology for Information Systems

We would like to set here some scheme for relating ontology of an information system outlined above to ontology of the associated mereological universe (again, we use term “ontology” here as a name for a system of notions only).

Given an information system (U, A), we introduce the material identity $=_U$ on U (understood as the identity of objects, e.g., identity of patients, signals, etc., witnessed by their names, times of receiving, etc.), and we let for $x, y\varepsilon U$,

$$x\mu_1 y \Leftrightarrow x =_U y. \tag{26}$$

Thus the notion of an *element*, corresponding to the rough inclusion μ by means of (23) coincides with $=_U$ on atomic objects; we will denote with the symbol el_U this notion of an element. We extend el_U to pairs $x \in g$, where $x\varepsilon U, g\varepsilon G_U$. The corresponding notion of a class, Cls_U coincides with the notion of a set. The mereological notion of a subset,

$$x sub_U y \Leftrightarrow \forall z.(z el_U x \Rightarrow z el_U y), \tag{27}$$

coincides with set inclusion $\subseteq$. Thus the mereological complement x^c (defined as the class of all objects that do not have any object as a common part with x) coincides with the set complement $U \setminus x$.

For each $x \varepsilon U$, the set,

$$Inf(x) = \{(a, a(x)) : a \in A\}, \tag{28}$$

will be defined, called the *information set of* x; the universe,

$$INF(U) = \{Inf(x) : x \varepsilon U\}, \tag{29}$$

will be called the *information universe* of the information system (U, A). We let

$$INF : U \to INF(U), INF(x) = Inf(x) \text{ for } x \varepsilon U. \tag{30}$$

Given a mereological structure on $INF(U)$, we denote with el_{INF}, Cls_{INF}, $=_{INF}$, respectively, the relation of being an element, a class, and the identity on $INF(U)$.

For $g \varepsilon G_U$, we let,

$$INF(g) =_{INF} Cls_{INF}(Inf(y) : y \varepsilon U, y el_U g). \tag{31}$$

We introduce names $INF(U), G_{INF}$ with $x \varepsilon INF(U)$ meaning x to be an atomic object of the form $Inf(a)$ with $a \varepsilon U$, and $x \varepsilon G_{INF}$ meaning x to be a class of atomic objects, i.e., $x = Cls_{INF}(\Phi | INF(U))$ for some non–vacuous property Φ with $\Phi | INF(U)$ being its restriction to INF–atomic objects, and non–vacuous on those objects.

Rough inclusions we define below, are defined on the universe $INF(U)$. As a result of operations in this universe, some constructs are obtained that we interpret in the universe U.

To this end, we define the semantic operator [.], which to any object g over the universe $INF(U)$ (be it an atom or a class) assigns the class $[g] = Cls_U(x \varepsilon U : Inf(x) el_{INF} g)$.

It is obvious that $x el_U [INF(x)]$; we state a property satisfied by [.], viz.,

$$g el_{INF} h \Rightarrow [g] el_U [h], \tag{32}$$

that follows directly from definitions and the transitivity of el_{INF}.

3 Examples

We consider an information system (U, A) and we define two basic examples of rough inclusions on $INF(U)$, that would be convenient in the sequel.

We define rough inclusions incrementally, invoking ontology introduced above. Both inclusions defined below satisfy the assumption that el_{INF} coincides with $=_{INF}$ on atomic objects. Thus in both cases, mereological classes coincide with sets; this may simplify the analysis for the reader. However, our analysis in general does not require this assumption.

3.1 The Menger Rough Inclusion

Given (U, A), hence, the information universe $INF(U)$, for x, y with $x\varepsilon INF(U)$, $y\varepsilon INF(U)$, where $x =_{INF} Inf(u), y =_{INF} Inf(v)$ with $u, v\varepsilon U$, we define,

$$DIS_A(x,y) = \{a \in A : a(u) \neq a(v)\}, \tag{33}$$

and then we let $y\mu_r^M x$ if and only if

$$e^{(-\Sigma_{a\in DIS_A(x,y)} w_a)} \geq r, \tag{34}$$

where $w_a \in (0, \infty)$ is a weight associated with the attribute a. Then μ^M does satisfy (18)–(25) with $x, y, z\varepsilon INF(U)$. For $x, y, z\varepsilon INF(U)$, the rule,

$$\frac{y\mu_r x, x\mu_s z}{y\mu_{r\cdot s} z}, \tag{35}$$

holds.

To justify our claim (35), we may observe that $DIS(y,z) \subseteq DIS(y,x) \cup DIS(x,z)$, which implies by (34) that $y\mu_r^M x, x\mu_s^M z$ imply $y\mu_{r\cdot s}^M z$, i.e., (35) holds.

It is a matter of straightforward checking that μ^M satisfies (18)–(25).

We now extend μ^M over pairs of the form x, g, where $x\varepsilon INF(U)$, $g\varepsilon G_{INF}$. We define μ^M in this case as follows:

$$x\mu_r^M g \Leftrightarrow \exists y.y\varepsilon U \wedge INF(y)el_{INF}g \wedge x\mu_r^M INF(y). \tag{36}$$

Then, the rough inclusion μ^M on pairs x, y or x, g, where $x, y\varepsilon INF(U)$, $g\varepsilon G_{INF}$, does satisfy (18)–(25).

Indeed, (18) and (25) are obviously true. For (23), $x\mu_1^M g$ implies that there exists $y\varepsilon INF(U), yel_{INF}g$ with $x\mu_1^M y$ hence $xel_{INF}y$ hence $x =_{INF} y$ so that $xel_{INF}g$. (24) follows on similar lines.

Finally, we define μ^M on pairs of the form g, h, with $g\varepsilon G_{INF}, h\varepsilon G_{INF}$, by means of the formula,

$$g\mu_r^M h \Leftrightarrow \forall x\varepsilon INF(U).[xel_{INF}g \Rightarrow \exists y\varepsilon INF(U).(yel_{INF}h \wedge x\mu_r^M y)]. \tag{37}$$

It is true that the extended by (37) most general form of μ^M does satisfy (18)–(25).

Indeed, (18) and (25) hold obviously; proof of (23) is as follows:
for $g, h\varepsilon G_{INF}$,

$$g\mu_1^M h, \tag{38}$$

if and only if

$$\forall x\varepsilon INF(U).[xel_{INF}g \Rightarrow \exists y\varepsilon INF(U).yel_{INF}h \wedge x\mu_1^M y], \tag{39}$$

if and only if

$$\forall x\varepsilon INF(U).[xel_{INF}g \Rightarrow \exists y\varepsilon INF(U).yel_{INF}h \wedge x =_{INF} y], \tag{40}$$

if and only if

$$\forall x \varepsilon INF(U).(x el_{INF} g \Rightarrow x el_{INF} h). \tag{41}$$

Now, from (41), the conclusion $g el_{INF} h$ follows by (21).

Proof for (24) goes on parallel lines: given $g\mu_1^M h$ and $k\mu_r^M g$, for any $x\varepsilon INF(U)$, where $x el_{INF} k$, there exist $y\varepsilon INF(U), y el_{INF} g$ with $x\mu_r^M y$; similarly, we find $z\varepsilon INF(U), z el_{INF} h$ with $y\mu_1^M z$. It follows by (35) that $x\mu_{r\cdot 1}^M z$ hence $x\mu_r^M z$ and thus, x being arbitrary, $k\mu_r^M h$.

A similar argument will show that the extended by (37) rough inclusion μ^M does satisfy the transitivity rule,

$$\frac{k\mu_r^M h, h\mu_s^M g}{k\mu_{r\cdot s}^M g}. \tag{42}$$

To prove (42), consider $k, h, g\varepsilon G_{INF}$; given $x\varepsilon INF(U), x el_{INF} k$, we find y, $z\varepsilon INF(U)$, $y el_{INF} h, z el_{INF} g$ with $x\mu_r^M y, y\mu_s^M z$, hence, by (35) we have $x\mu_{r\cdot s}^M z$ implying by arbitrariness of x that $k\mu_{r\cdot s}^M g$.

3.2 The Łukasiewicz Rough Inclusion

In notation of (33), we define a rough inclusion μ^L. Following the pattern of the discussion in (34), (35), the rough inclusion μ^L will be defined as follows.

For $x, y\varepsilon INF(U)$,

$$y\mu_r^L x \Leftrightarrow 1 - \frac{|DIS_A(x,y)|}{|A|} \geq r. \tag{43}$$

For $g, h\varepsilon \{INF(U), G_{INF}\}$:

$$(g\mu_r^L h) \Leftrightarrow \{[\forall x\varepsilon INF(U).(x el_{INF} g \Rightarrow \exists y\varepsilon INF(U).(y el_{INF} h \wedge x\mu_r^L y)]\}. \tag{44}$$

Following the lines of proofs of (18)–(25) in case of the Menger rough inclusion, and of (42) in which the rough inclusion μ^M is replaced with the rough inclusion μ^L, we obtain that *the Łukasiewicz rough inclusion μ^L does satisfy* (18)–(25) *with* $x, y, z\varepsilon\{INF(U), G_{INF}\}$, and the following rule,

$$\frac{k\mu_r^L g, g\mu_s^L h}{k\mu_{\otimes(r,s)}^L h}, \tag{45}$$

holds, where $\otimes(r,s)$ is the Łukasiewicz functor (10).

As with the Menger rough inclusion, it is sufficient to verify (18)–(25) with $x, y, z\varepsilon INF(U)$ which is straightforward and then to follow general lines outlined above.

The transitivity rule (45) deserves a proof. It is sufficient to assume $x\mu_r^L y$, $y\mu_s^L z, x\mu_t^L z$ with $x, y, z\varepsilon INF(U)$.

As $DIS(x,z) \subseteq DIS(x,y) \cup DIS(y,z)$, substituting into (43),

$$1 - t = \frac{|DIS(x,z)|}{|A|} \leq \frac{|DIS(x,y)|}{|A|} + \frac{|DIS(y,z)|}{|A|} = (1-r) + (1-s), \quad (46)$$

hence $t \geq r+s-1$ and as t is non–negative we have finally $t \geq max\{0, r+s-1\} = \otimes(r,s)$.

The proof in the general case goes on lines of the proof in case of Menger's rough inclusion.

We add a new rule about rough inclusions, which is satisfied by Menger as well as Łukasiewicz rough inclusions with $f = Prod$, $\otimes$ respectively.

The *f*-Transitivity Property

$$\frac{x\mu_r y, y\mu_s z}{x\mu_{f(r,s)} z} \quad (47)$$

holds with a t–norm f.

We say that μ is an *f–rough inclusion.*

Note, that it follows from general properties of t–norms, that for each t–norm, T, there exists a rough inclusion μ^T that satisfies the T–transitivity rule, i.e.,

$$\frac{k\mu_r^T g, g\mu_s^T h}{k\mu_{T(r,s)}^T h}, \quad (48)$$

see [22], Prop. 13.

Finally, we state for the record the obvious property of Menger as well as Łukasiewicz rough inclusions, viz. symmetry, of which no explicit use was made as of yet.

The Symmetry Property

$$x\mu_r y \Leftrightarrow y\mu_r x, \quad (49)$$

holds for $\mu \in \{\mu^M, \mu^L\}$.

3.3 Remarks on the Notion of an Element Induced by a Rough Inclusion

In cases of Menger as well as Łukasiewicz rough inclusions, formulas defining μ are employing cardinalities of sets $DIS_A(x,y)$; in consequence, the notion of an element el_{INF} induced by either of μ^M, μ^L coincides with the notion of being indiscernible: $xel_{INF}y$ if and only if $IND_A(u(x), u(y))$ if and only if $u(x) =_U u(y)$, where $x = INF(u(x)), y = INF(u(y))$.

At the cost of introducing an additional structure into value sets V_a, where $a \in A$, we may generate a richer el_{INF} relation. For instance, we may assume preference relations $<_a$ in sets V_a, i.e., orderings on value sets V_a, studied by some authors, e.g., Greco et al., see [6]. Then, we may let,

$$DIS_A^<(x,y) = \{a \in A : a(x) < a(y)\}, \quad (50)$$

and,

$$x\mu_r^{L,<}y \Leftrightarrow 1 - \frac{|DIS_A^{<}(x,y)|}{|A|} \geq r. \tag{51}$$

Alternatively, we may introduce the set

$$DIS_A^{\geq}(x,y) = \{a \in A : a(x) \geq a(y)\}, \tag{52}$$

hence,

$$DIS_A^{\geq}(x,y) = A \setminus DIS_A^{<}(x,y), \tag{53}$$

and, in consequence,

$$x\mu_r^{L,<}y \Leftrightarrow |DIS_A^{\geq}(x,y)| \geq r \cdot |A|. \tag{54}$$

Thus, the notion an element $el^{<}_{INF}$ is as follows: $xel^{<}_{INF}y$ if and only if $a(x) \geq a(y)$ for each a in A.

It is straightforward to verify that $\mu^{L,<}$ is a rough inclusion; only (23) may require a comment. In this case, we consider x, y, z with $x\mu_1^{L,<}y$ and $z\mu_r^{L,<}x$.

We have: $DIS_A^{\geq}(x,y) = A$, i.e., $a(x) \geq a(y)$ for every a in A, and

$$|DIS_A^{\geq}(z,x)| \geq r \cdot |A|.$$

As $a(z) \geq a(x)$ implies $a(z) \geq a(y)$, it follows that $DIS_A^{\geq}(z,x) \subseteq DIS_A^{\geq}(z,y)$, hence, $|DIS_A^{\geq}(z,y)| \geq |DIS_A^{\geq}(z,x)| \geq r \cdot |A|$, i.e., $z\mu_r^{L,<}y$.

In an analogous way, we may define the corresponding Menger rough inclusion $\mu_r^{M,<}$, i.e.,

$$x\mu_r^{M,<}y \Leftrightarrow e^{-\Sigma_{a \in DIS_A^{\geq}(x,y)} w_a} \geq r. \tag{55}$$

Let us observe that

$$e^{-\Sigma_{a \in DIS_A(x,y)} w_a)} = 1 - \frac{\Sigma_{a \in DIS_A(x,y)} w_a}{1!} + \frac{(\Sigma_{a \in DIS_A(x,y)} w_a)^2}{2!} - ..., \tag{56}$$

and, assuming that the average value of $|DIS_A(x,y)|$ is $\frac{|A|}{2}$, letting in (56) $w_a = \frac{1}{|A|}$, for each $a \in A$, we obtain that

$$e^{-\Sigma_{a \in DIS_A(x,y)} w_a)} = 1 - \frac{|DIS_A(x,y)|}{|A|}, \tag{57}$$

with an error of order 0.06. We may, therefore, base examples that follow on the Łukasiewicz rough inclusion as a fairly close estimate to the Menger rough inclusion.

We consider the information system $\mathcal{A}$ in Table 1.

For the information system $\mathcal{A}$, we may calculate values of the Łukasiewicz rough inclusion in Table 2; as μ^L is symmetric, we show only the upper triangle of values.

As already observed, the relation of *element* is equal to the identity $=_U$.

Table 1. The information system $\mathcal{A}$.

U	a_1	a_2	a_3	a_4
x_1	1	1	1	2
x_2	1	0	1	0
x_3	2	0	1	1
x_4	3	2	1	0
x_5	3	1	1	0
x_6	3	2	1	2
x_7	1	2	0	1
x_8	2	0	0	2

Table 2. μ^L for Table 2.

U	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1	0.5	0.25	0.25	0.5	0.5	0.25	0.25
x_2	*	1	0.5	0.5	0.5	0.25	0.25	0.25
x_3	*	*	1	0.25	0.25	0.25	0.25	0.5
x_4	*	*	*	1	0.75	0.75	0.25	0.0
x_5	*	*	*	*	1	0.5	0	0
x_6	*	*	*	*	*	1	0.25	0.25
x_7	*	*	*	*	*	*	1	0.25
x_8	*	*	*	*	*	*	*	1

Table 3. $\mu^{<,L}$ for Table 2.

U	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1	1	0.75	0.5	0.75	0.5	0.75	0.75
x_2	0.5	1	0.5	0.5	0.5	0.25	0.5	0.5
x_3	0.5	1	1	0.5	0.5	0.25	0.75	0.75
x_4	0.75	1	0.75	1	1	0.75	0.75	0.75
x_5	0.75	1	0.75	0.75	1	0.5	0.5	0.75
x_6	1	1	1	1	1	1	1	1
x_7	0.5	0.75	0.5	0.5	0.5	0.25	1	0.5
x_8	0.5	0.75	0.75	0.25	0.25	0.25	0.75	1

Taking the natural ordering of integers as the preference ordering in all sets V_a for $a = a_1, a_2, a_3, a_4$, we induce the rough inclusion $\mu^{L,<}$ in the universe of the system $\mathcal{A}$, whose values are shown in Table 3.

As we see, the rough inclusion $\mu^{L,<}$ is not symmetric, and we may expect a different structure of granules, as we pass to the subject of knowledge granulation.

3.4 Granulation of Knowledge

The class operator Cls may be recalled now. We make use of it in defining a notion of a granule in the rough mereological universe. Granular computing (GC) as a paradigm has been proposed by Lotfi Zadeh [28] and it has been studied intensively by rough and fuzzy set theorists, see [15], [12]. According to an informal, intuitive, definition, due to Zadeh, a granule is a collection (abstracting from the precise meaning of the term) of objects bound together by similarity, closeness, or some other constraints alike. Here, we construct granules on the basis of rough mereological similarity of objects. Due to regularity of assumptions about rough inclusions, we are able to obtain also some regular properties of granules.

4 Rough Mereological Granulation

We assume a rough inclusion μ on a mereological universe (U, el). For given $r < 1$ and $x \varepsilon U$, we let,

$$g_r(x) = Cls(\Psi_r), \tag{58}$$

where,

$$\Psi_r(y) \Leftrightarrow y\mu_r x. \tag{59}$$

The class $g_r(x)$ collects all atomic objects satisfying the class definition with the concept Ψ_r.

We will call the class $g_r(x)$ the *r–granule* about x; it may be interpreted as a neighborhood of x of radius r. We may also regard the formula $y\mu_r x$ as stating *similarity* of y to x (to degree r). We do not discuss here the problem of representation of granules; in general one may apply sets as the underlying representation structure. It will turn out, that in case of Menger as well as Łukasiewicz rough inclusions, granules may be represented as sets of objects satisfying Ψ_r.

From (18)–(25), the following general properties of granulation operators g_r may be deduced.

$$y\mu_r x \Rightarrow yel g_r(x), \tag{60}$$

$$x\mu_r y \wedge yelz \Rightarrow xel g_r(z), \tag{61}$$

$$\forall z.[zely \Rightarrow \exists w, q.(welz \wedge welq \wedge q\mu_r(x))] \Rightarrow yel g_r(x), \tag{62}$$

$$yel g_r(x) \wedge zely \Rightarrow zel g_r(x), \tag{63}$$

$$s \leq r \Rightarrow g_r(x) el g_s(x). \tag{64}$$

Property (60) follows by definition of g_r and class definition (19, 20). Property (61) is implied by (23) and (60). Property (62) follows by (21). Property (63) follows by transitivity of el. Property (64) is a consequence of (25).

Let us observe that in case of DIS_A – induced rough inclusions, $g_1(x) = x$ is the class of elements of x hence x itself. By (60–64), the system $\{g_r(x) : x \in U, r \in [0,1]\}$ is a neighborhood system for a weak topology on the universe U.

In the matter of example, we consider the information system of Table 1 along with values of rough inclusions $\mu^L, \mu^{L,<}$ given, respectively, in Tables 2, 3. Admitting $r = 0.5$, we list below granules of radii 0.5 about objects $x_1 - x_8$ in both cases. We denote with the symbol $g_i, g_i^<$, respectively, the granule $g_{0.5}(x_i)$ defined by $\mu_L, \mu_{L,<}$, respectively.

We have,

1. $g_1 = \{x_1, x_2, x_5, x_6\}$,
2. $g_2 = \{x_1, x_2, x_3, x_4, x_5\}$,
3. $g_3 = \{x_2, x_3, x_8\}$,
4. $g_4 = \{x_2, x_4, x_5, x_6\}$,
5. $g_5 = \{x_1, x_2, x_4, x_5, x_6\}$,
6. $g_6 = \{x_1, x_4, x_5, x_6\}$,
7. $g_7 = \{x_7\}$,
8. $g_8 = \{x_3, x_8\}$,

what provides an intricate relationship among granules: $_1, g_4, g_6 \subseteq g_5$, $g_8 \subseteq g_2$, g_2, g_5 incomparable by inclusion, g_7 isolated. We may contrast this picture with that for $\mu^{L,<}$.

1. $g_1^< = g_4^< = g_5^< = g_6^< = U$,
2. $g_2^< = g_3^< = g_7^< = U \setminus \{x_6\}$,
3. $g_8^< = \{x_1, x_2, x_3, x_7, x_8\}$,

providing a nested sequence of three distinct granules.

4.1 Granulation via the Menger Rough Inclusion

We now consider the Menger rough inclusion μ^M in which case the following additional properties of granules may be observed. The r-granule about x induced by μ^M will be denoted with the symbol g_r^M.

The following are additional properties of Menger granules,

$$x\varepsilon INF(U) \wedge x el_{INF} g_r^M(y) \Leftrightarrow x\mu_r^M y, \tag{65}$$

$$x\varepsilon INF(U) \wedge x el_{INF} g_r^M(y) \Rightarrow \forall t \in [0,1].g_t^M(x) el_{INF} g_{t\cdot r}^M(y). \tag{66}$$

For (65), assume that $x el_{INF} g_r^M(y)$ for $x\varepsilon INF(U)$; then, by class definition (19, 20), we find w, q such that $w el_{INF} x, w el_{INF} q, q\mu_r^M y$. Thus $w\mu_{1\cdot r}^M y$, hence $w\mu_r^M y$, and again, by $x\mu_1^M w$, it follows that $x\mu_r^M y$. Proof of (66) follows on the same lines.

4.2 Granulation via the Łukasiewicz Rough Inclusion

With the Łukasiewicz rough inclusion, analogous observations follow whose proofs are carried on the same lines. The following are additional properties of Łukasiewicz granules,

$$x\varepsilon INF(U) \wedge x el_{INF} g_r^L(y) \Leftrightarrow x\mu_r^L y, \tag{67}$$

$$x\varepsilon INF(U) \wedge x el_{INF} g_r^L(y) \Rightarrow \forall t \in [0,1].g_t^L(x) el_{INF} g_{\otimes(t,r)}^L(y). \tag{68}$$

We may sum up more suggestively (66, 68), viz., for $\mu \in \{\mu^M, \mu^L\}$ and $x\varepsilon INF(U)$, and sufficiently small positive ε,

$$x el_{INF} g_r(y) \Rightarrow g_{1-\varepsilon}(x) el_{INF} g_{r-\varepsilon}(y). \tag{69}$$

In the sequel, we discuss schemes for computing with granules of knowledge/computing with words leading to networks of units organized similarly to neural networks but performing their computations in different manner. Hybrid rough-neural computing schemes are recently given much attention, see [15]. Here, we propose a look at the schemes based purely on agents equipped with rough inclusions. Differentiability of the Menger type rough inclusions make them susceptible to learning by back-propagation, a subject we discuss in the final part of the following section.

5 Rough Mereological Granular Computing

We define an intelligent unit modelled on a classical perceptron [1] and then we develop the notion of a network of intelligent units of this type. The scheme presented here is a modification of the scheme proposed by Polkowski and Skowron in [22].

5.1 Rough Mereological Perceptron

We exhibit the structure of a *rough mereological perceptron* (RMP). It consists of an intelligent unit denoted ia. The input to ia is a finite set of connections $Link_{ia,in} = \{link_1, ..., link_m\}$; each $link_j$ has as the source the information universe $INF(U_j)$, of an information system $\mathcal{A}_j = (U_j, A_j)$, endowed with a rough inclusion μ_r^j. The output to ia is a connection $Link_{ia,out}$ to the information universe $INF(U_{ia})$, of an information system $\mathcal{A}_{ia} = (U_{ia}, A_{ia})$, equipped with the rough inclusion μ^{ia}.

The operation (function) realized in RMP is denoted with O_{ia}; thus, for every tuple $< x_1, ..., x_m >$, where $x_i \in INF(U_i)$, the object $x = O_{ia}(x_1, ..., x_m) \in INF(U_{ia})$.

In each $INF(U_j)$ as well as in $INF(U_{ia})$, finite sets T_j, T_{ia} are selected, with the properties that,

$$t \in t_{ia} \Rightarrow \exists t_1 \in T_1, ..., t_m \in T_m . t = O_{ia}(t_1, .., t_m), \tag{70}$$

and,

$$\forall i, t_i \in T_i . \exists t \in T_{ia} . t = O_{ia}(t_1, ..., t_m). \tag{71}$$

When $t = O_{ia}(t_1, ..., t_m)$, we say that $\sigma =< t_1, ..., t_m, t >$ is an *admissible set of references*. The set of all admissible reference sets is denoted by Σ.

The operation of an RMP may be expressed in terms of the mapping ω_{ia} defined as follows.

For $\sigma =< t_1, ..., t_m, t >$, $r_1, ..., r_m \in [0, 1]$, $x = O_{ia}(x_1, ..., x_m)$, and $x_i \varepsilon INF(U_i)$ for $i = 1, 2, ..., m$, the mapping,

$$\omega_{ia} : \Sigma \times [0, 1]^m \to [0, 1], \tag{72}$$

is given by

$$\omega_{ia}(\sigma, r_1, .., r_m) = r = min\{s : x \mu_s^{ia} t\}, \tag{73}$$

whenever $x_i \mu_{r_i}^i t_i$ for $i = 1, 2, ..., m$.

5.2 Granular Computations

The mapping ω_{ia} may be factored through granule operator, i.e., (73) may be expressed as follows:

$$\omega_{ia}^g(\sigma, g_{r_1}(t_1), ..., g_{r_m}(t_m)) = g_{\omega_{ia}(\sigma, r_1, ..., r_m)}(t), \tag{74}$$

defining the factored mapping ω^g on granules.

Let us observe that the acting of RMP may as well be described as that of a *granular controller*, viz., the functor ω^g may be described via a decision algorithm consisting of rules of the form,

$$\mathbf{i}f\ g_{r_1}(t_1)\ \mathbf{a}nd\ g_{r_2}(t_2)\ \mathbf{a}nd\ \ldots\ \mathbf{a}nd\ g_{r_m}(t_m)\ \mathbf{t}hen\ g_r(t), \tag{75}$$

with $r = \omega_{ia}(\sigma, r_1, ..., r_m)$, where $\sigma =< t_1, ..., t_m, t >$.

It is worth noticing that the mapping ω_{ia} is defined from given information systems $\mathcal{A}_j, \mathcal{A}_{ia}$ and it is not subject to any arbitrary choice.

Composition of RMP's involves a composition of the corresponding mappings ω, viz., given $RMP_1, \ldots, RMP_k$, RMP with links to RMP being outputs from $RMP_1, \ldots, RMP_k$, each RMP_j having inputs $Link_j = \{link^j_1, . .., link^j_{k_j}\}$, $m = \Sigma^k_{j=1} k_j$, the composition,

$$IA = RMP \circ (RMP_1, RMP_2, ..., RMP_k), \tag{76}$$

of m inputs, does satisfy the formula

$$\omega_{IA} = \omega_{RMP} \circ (\omega_{RMP_1}, ..., \omega_{RMP_k}), \tag{77}$$

under the tacit condition that admissible sets of references are composed as well. Thus RMP's may be connected in networks subject to standard procedures, e.g., learning by back-propagation.

When learning by back-propagation (see [1], [20] for a detailed analysis), the Menger rough inclusion, or its modifications, e.g., the *gaussian rough inclusion*,

$$x\mu^G_r y \Leftrightarrow e^{-(\Sigma_{a \in DIS_A(x,y)} w_a)^2} \geq r, \tag{78}$$

are useful as they are defined by means of a differentiable function f, whose gradient is of the form,

$$\frac{\partial f}{\partial w} = f \cdot (-2 \cdot \sum w_a), \tag{79}$$

in case of μ^G.

5.3 Elementary Teams of Rough Mereological Perceptrons with the Menger Rough Inclusion

We consider a *rough mereological perceptron* (RMP), as defined in Sect. 5.1. It consists of an intelligent unit denoted ia whose input is a finite tuple $\overline{x} =< x_1, x_2, \ldots, x_k >$ of objects, see Sect. 5.1. We endow ia with the Menger rough inclusion $\mu^{M,ia}$ at ia.

The unit ia is also equipped with a set of *target objects* $T_{ia} \subset INF(U_{ia})$. Each object in T_{ia} represents a class of indiscernibility $IND_{A_{ia}}$ via the map INF.

The output of RMP is the granule of knowledge $res_{ia}(\overline{x}) = g_{r_{res}}(x)$ with the property that

$$r_{res} = max\{r : \exists y \in T_{ia}.O_{ia}(\overline{x})\mu^{M,ia}_r y\}, \tag{80}$$

i.e., we essentially classify $\overline{x}$ as the exact concept being the class of indiscernibility classes as close to the indiscernibility class of x as the closest target class, i.e., in degree r_{res}. Let us observe that this classification depends on the weight system $\{w_a : a \in A_{ia}\}$ chosen. We may also observe that we could also present the result of computation $res_{ia}(\overline{x})$ as the tuple $< g_{r_i}(t_i) : t_i \in T_{ia} >$ where $r_i = sup\{r : O_{ia}(\overline{x})\mu_r^{M,ia} t_i\}$ for each $i \leq |T_{ia}|$.

5.4 Rough Mereological Network

Now, we may consider a network of intelligent agents (NIA) organized in the manner of a feed-forward neural network, viz., we single out the following components in NIA.

1. The set *INPUT* of intelligent agents (*ia*'s) constituting the input layer of NIA. Each intelligent unit *ia* in *INPUT* acts as an RMP: it receives a tuple $\overline{x}_{ia}$ of objects from the stock of primitive objects (signals, etc.); it assembles $\overline{x}_{ia}$ into the object $O_{ia}(\overline{x}_{ia})$ in U_{ia}; it classifies $O_{ia}(\overline{x}_{ia})$ with respect to its target concepts in T_{ia} and outputs the result $res_{ia}(\overline{x}_{ia})$.
2. Sets $LEV_1, \ldots, LEV_k$ constituting consecutive inner layers of NIA with each ia in $\bigcup_i LEV_i$ acting as above.
3. The output to RMP denoted ia_{out} acting as each ia except that its computation result is the collective computation result of the whole network.

Although, in general, the structure of neural networks is modelled on directed acyclic graphs, we assume for simplicity, that NIA is ordered by a relation ρ into a tree. Thus ia_{out} is the root of the tree, $INPUT$ is the leaf set of the tree, and each LEV_i constitutes the corresponding level of pairwise non–communicating units in the tree.

Elementary Computations. To analyze computation mechanisms in NIA, we begin with *elementary computations* performed by each subtree of the form $nia = \{ia, ia_0, \ldots, ia_m\}$ with $ia_j \rho ia$ for each $j \leq m$, i.e., ia is the root unit in nia, and $ia_0, \ldots, ia_m$ are its daughter units.

We make one essential assumption about NIA, viz., we presume that after preliminary training, the target sets T_{ia}, T_{ia_j} have been coordinated, i.e.,

1. for each $t \in T_{ia}$, there exist $t_1 \in T_{ia_1}, \ldots, t_m \in T_{ia_m}$ such that $t = O_{ia}(t_1, \ldots, t_m)$;
2. each $t_j \in T_{ia_j}$ can be completed by $t_1 \in T_{ia_1}, \ldots, t_{j-1} \in T_{ia_{j-1}}, t_{j+1} \in T_{ia_{j+1}}, \ldots$, $t_m \in T_{ia_m}$ such that $t = O_{ia}(t_1, \ldots, t_m) \in T_{ia}$.

On the basis of these assumptions, we can describe elementary computations in terms of target sets. To this end, we introduce notions of *propagating functors* relative to sets of target concepts. The notion of a propagating functor was introduced in [22], and here we adopt it in a slightly changed form.

We will say that a set $\sigma = \{t, t_1, \ldots, t_m\}$ of target concepts, where $t \in T_{ia}, t_1 \in T_{ia_1}, \ldots, t_m \in T_{ia_m}$ is *admissible* when $t = O_{ia}(t_1, \ldots, t_m)$. For an admissible $\sigma = \{t, t_1, \ldots, t_m\}$ of target concepts, where $t \in T_{ia}, t_1 \in T_{ia_1}, \ldots, t_m \in T_{ia_m}$, the propagating functor $\Phi_{nia,\sigma}$ is defined as follows.

For $\overline{x} =< x_1 \in INF(U_{ia_1}), \ldots, x_m \in INF(U_{ia_m}) >$, we denote by $\overline{r}$ the tuple $< r_1, \ldots, r_m > \in \mathbf{R}^m$ defined by

$$\forall i.r_i = sup\{r : x_i \mu_r^{M,ia_i} t_i\}, \tag{81}$$

and we let

$$\Phi_{nia,\sigma}(\overline{r}) = sup\{s : O_{ia}(x_1, \ldots, x_m)\mu_s^{M,ia} t\}. \tag{82}$$

In terms of functors $\Phi_{nia,\sigma}$, reasoning and computations are carried out in NIA (see [22], [20] for details). As induced by the function f, $\Phi_{nia,\sigma}$ is piece–wise differentiable, allowing for back–propagation based learning in NIA (see [20]).

Computing with Words. The paradigm of computing with words [29] assumes that syntax of the computing mechanism is given as that of a subset of natural language, whereas the semantics is given in a formal computing mechanism involving numbers.

Let us comment briefly on how this idea may be implemented, exemplarily, in an RMP. Assume, there is given a set $\mathcal{N}$ of noun phrases $\{n_1, n_2, ..., n_m, n\}$ corresponding to information system universes $U_1, ..., U_m$. A set $\mathcal{ADJ}$ of adjective phrases is also given, and to each $\sigma \in \Sigma$, a set $adj_1, .., adj_m, adj$ is assigned.

Then the decision rule (75) may be expressed in the form,

$$\mathbf{i}f\ n_1\ \mathbf{is}\ adj_1\ \mathbf{a}nd\ \ldots \mathbf{a}nd\ n_m\ \mathbf{i}s\ adj_m\ \mathbf{t}hen\ n\ \mathbf{i}s\ adj. \tag{83}$$

The semantics of (83) is expressed in the form of (75). The reader will observe that (83) is similar in form to decision rules of a fuzzy controller, while the semantics is distinct.

Composition of RMP's as above is reflected in compositions of rules of the form (83) with semantics expressed by composed mappings ω.

In the light of analysis that begins in the next section, we may call the computation model presented above, a rough–fuzzy–neurocomputing model, see [15] for discussions of this topic.

6 Rough Set Approximations and Fuzzy Partitions and Equivalences Induced by Rough Inclusions

We assume that a mereological universe U is given with a rough inclusion μ of which we assume symmetry (49) and an f–transitivity property (47) with a t–norm f. We will construct in this setting rough respectively fuzzy universes derived from the rough inclusion in question, and represented by rough set approximations and by partitions and equivalence relations in the sense of fuzzy set theory.

In rough as well as fuzzy set–theoretic literature, many authors have devoted their attention to the problem of mutual relationships between the two theories and, in consequence, some notions of a rough –fuzzy set as well as a fuzzy–rough set have emerged. The reader will consult [2], [3], [5], [8], [9], [10], [25], as well as many chapters in this volume.

6.1 Rough Environment from Rough Inclusions

We first give an abstract description of our ideas in the rough case and then we explicate them in case of information systems and Menger/Łukasiewicz rough inclusions. We will here discern between universes $U, INF(U)$, which will bear on notation.

Assume a rough inclusion μ on a mereological universe $(INF(U), el_{INF})$ and a non–vacuous property M on classes over $INF(U)$; given a class x over $INF(U)$, we define its *lower* M, μ*–approximation* $\underline{x}^{\mu}_{M}$ as follows,

$$\underline{x}^{\mu}_{M} = Cls(g \varepsilon M : g \mu_1 x). \tag{84}$$

Then we have as a direct consequence of (21),

$$\underline{x^{\mu}}_{M}\ el\ x. \tag{85}$$

Indeed, for $y\ el_{INF} \underline{x^{\mu}}_{M}$ there exist w, q with

$$w el_{INF} y, w el_{INF} q, q \varepsilon M, q \mu_1 x, \tag{86}$$

hence, $q el_{INF} x$ by (18), and by the inference rule (21) it follows that,
$\underline{x^{\mu}}_{M}\ el_{INF} x$.

Again, applying the inference rule (21), we obtain,

$$\underline{x}^{\mu}_{M}\ =_{INF} \underline{\underline{x}^{\mu}_{M}}^{\mu}_{M}. \tag{87}$$

It is sufficient to verify that $\underline{x}^{\mu}_{M}\ el_{INF} \underline{\underline{x}^{\mu}_{M}}^{\mu}_{M}$, the converse being satisfied in virtue of (85). Given $y el_{INF} \underline{x}^{\mu}_{M}$, we have w, q with

$$w el_{INF} y, w el_{INF} q, q \varepsilon M, q \mu_1 x, \tag{88}$$

hence, $q el_{INF} \underline{x}^{\mu}_{M}$, i.e., $q \mu_1 \underline{x}^{\mu}_{M}$ i.e. $q el_{INF} \underline{\underline{x}^{\mu}_{M}}^{\mu}_{M}$.

It follows by (21) that $\underline{x}^{\mu}_{M}\ el_{INF} \underline{\underline{x}^{\mu}_{M}}^{\mu}_{M}$.

We will say that x is M*–exact* in case there exists a non–empty $M_0 \subseteq M$ such that $x =_{INF} Cls(M_0)$.

Then, if

$$x = \underline{x}^{\mu}_{M}, \tag{89}$$

then x is M–exact.

Indeed, if $x = \underline{x}^{\mu}_{M}$ then $x = Cls(g \varepsilon M : g \mu_1 x)$ so it suffices to let $M_0(g) \Leftrightarrow g \varepsilon M \wedge y \mu_1 x$.

The converse also holds, i.e., if x is M–exact, $x = Cls(M_0)$ with some $M_0 \subseteq M$ then,

$$x =_{INF} \underline{x}^{\mu}_{M}. \tag{90}$$

Indeed, given $yel_{INF}x$, we have w, q with

$$wel_{INF}y, wel_{INF}q, q\varepsilon M_0, qel_{INF}x, \tag{91}$$

hence, $qel_{INF}\underline{x}^{\mu}_{M}$, and (21) implies that $xel_{INF}\underline{x}^{\mu}_{M}$, so (90) follows by (85).

Now, we define the *upper* M, μ*–approximation* $\overline{x}^{\mu}_{M}$ by letting

$$\overline{x}^{\mu}_{M} = Cls(g\varepsilon M : \exists yel_{INF}x.y\mu_1 g). \tag{92}$$

This case requires some assumptions about M. We will say that M is a *covering* in case the condition holds,

$$\forall x\varepsilon INF(U).\ \exists g\varepsilon M.\ x\mu_1 g. \tag{93}$$

We say that M is a *partition* in case the following condition holds,

$$\forall h, g\varepsilon M.(h \neq_{INF} g \Rightarrow ext(h, g)), \tag{94}$$

where

$$ext(h, g) \Leftrightarrow \neg[\exists z\varepsilon INF(U).\ (zel_{INF}h \wedge zel_{INF}g)]. \tag{95}$$

It is true that if M is a covering, then

$$xel_{INF}\overline{x}^{\mu}_{M}. \tag{96}$$

For $yel_{INF}x$, there exists $g\varepsilon M$ with $y\mu_1 g$ and then $yel_{INF}g, gel_{INF}\overline{x}^{\mu}_{M}$; by (21), $xel_{INF}\overline{x}^{\mu}_{M}$.

If M is a covering, then

$$\overline{x}^{\mu}_{M} =_{INF} \overline{\overline{x}^{\mu}_{M}}^{\mu}_{M}. \tag{97}$$

It suffices by (96) to verify the element inclusion of the right–hand side into the left–hand side. So, let $yel_{INF}\overline{\overline{x}^{\mu}_{M}}^{\mu}_{M}$. Thus there exist w, q with

$$wel_{INF}y, wel_{INF}q, qel_{INF}\overline{x}^{\mu}_{M}, \tag{98}$$

hence, there exist t, r, z with

$$tel_{INF}w, tel_{INF}r, r\varepsilon Mzel_{INF}x, zel_{INF}r, \tag{99}$$

and thus $rel_{INF}\overline{x}^{\mu}_{M}$, and as $tel_{INF}y$ by (98), (99), and transitivity of el_{INF}, the entities y, t, r do witness by (21) that $\overline{\overline{x}^{\mu}_{M}}^{\mu}_{M}el_{INF}\overline{x}^{\mu}_{M}$.

The question whether $x =_{INF} \overline{x}^{\mu}_{M}$ has a positive answer, too.

If M is a covering and a partition, and $x =_{INF} Cls(M_0)$ with some $\emptyset \neq M_0 \subseteq M$, then,

$$x =_{INF} \overline{x}^{\mu}_{M}. \tag{100}$$

It suffices to verify that $\overline{x}^{\mu}_{M} el_{INF} x$. To this end, we apply (21), and given $y el_{INF} \overline{x}^{\mu}_{M}$, we find w, q, z such that

$$w el_{INF} y, w el_{INF} q, q el_{INF} \overline{x}^{\mu}_{M}, q \varepsilon M, z el_{INF} x, z el_{INF} q. \tag{101}$$

As $x =_{INF} Cls(M_0)$, there exist u, r with

$$u el_{INF} z, u el_{INF} r, r \varepsilon M_0. \tag{102}$$

It follows that $u el_{INF} q, u el_{INF} r$ so $\neg ext(q, r)$ holds and thus by (95), $q =_{INF} r$. As $x =_{INF} Cls(M_0)$, $r el_{INF} x$ and from,

$$w el_{INF} r, w el_{INF} y, r el_{INF} x, \tag{103}$$

we infer by (21)) that $\overline{x}^{\mu}_{M} el_{INF} x$, concluding the proof.

In consequence of (89, 90, 100), when M is a covering and a partition, $\overline{x}^{\mu}_{M} = x = \underline{x}^{\mu}_{M}$ if and only if x is M–exact.

The above properties witness that rough mereological approximations share well–known [16] properties of rough set approximations.

We now should restore from rough mereological approximations $\underline{x}^{\mu}_{M}, \overline{x}^{\mu}_{M}$, rough set approximations $\underline{A}x, \overline{A}x$ over the universe U.

We first define the upper approximation. To this end, we make use of the semantic operator [.] subject to (32), and we let

$$\overline{x}^{M,\mu} = [\overline{INF(x)}^{\mu}_{M}]. \tag{104}$$

The following properties are true.

If M is a covering, then

$$x el_U \overline{x}^{M,\mu}. \tag{105}$$

If M is a covering, then

$$\overline{x}^{M,\mu} =_U \overline{\overline{x}^{M,\mu}}^{M,\mu}. \tag{106}$$

If M is a covering and a partition, then

$$x =_U \overline{x}^{M,\mu}, \tag{107}$$

whenever $x =_U [INF(x)]$ and $[INF(x)]$ is M–exact.

Property (105) follows from (32) and (96). Similarly, (106) follows from (32) and (97). For (107), if $INF(x)$ is M–exact, then

$INF(x) =_{INF} \overline{INF(x)}^{\mu}_{M}$ by (U3),

hence, $\overline{x}^{M,\mu} =_U [\overline{INF(x)}^{\mu}_{M}] =_U [INF(x)] =_U x$ by (32).

To define the lower approximation $\underline{x}_{M,\mu}$ to x, we make use of the upper approximation,

$$\underline{x}_{M,\mu} =_U (\overline{x^c}^{M,\mu})^c. \tag{108}$$

The following properties hold.

If M is a covering, then

$$\underline{x}_{M,\mu} el_U x. \tag{109}$$

If M is a covering, then

$$\underline{x}_{M,\mu} =_U \underline{\underline{x}_{M,\mu}}_{M,\mu}. \tag{110}$$

If $x =_U [INF(x)]$, $INF(x)]$ is M–exact, and M is a covering and a partition, then

$$x =_U \underline{x}_{M,\mu}. \tag{111}$$

Equation (109) holds as $x^c el_U \overline{x^c}^{M,\mu}$ by (105), hence

$$\underline{x}_{M,\mu} =_U (\overline{x^c}^{M,\mu})^c el_U x^{cc} =_U x.$$

Equation (110) results from (106), as

$$\underline{\underline{x}_{M,\mu}}_{M,\mu} =_U (\overline{(\underline{x}_{M,\mu})^c}^{M,\mu})^c =_U (\overline{(\overline{x^c}^{M,\mu})^{cc}}^{M,\mu})^c =_U$$

$$(\overline{(\overline{x^c}^{M,\mu})}^{M,\mu})^c =_U (\overline{x^c}^{M,\mu})^c =_U \underline{x}_{M,\mu}.$$

Equation (111) follows from (107), as $x^c =_U [INF(x^c)]$ and $INF(x^c)$ is M–exact hence $x^c =_U \overline{x^c}^{M,\mu}$ and thus $x =_U x^{cc} =_U (\overline{x^c}^{M,\mu})^c = \underline{x}_{M,\mu}$.

6.2 The Case of Menger, Łukasiewicz Rough Inclusions

We insert here a short interlude on the approximation theme using the Menger or the Łukasiewicz rough inclusions to define granules denoted in this case with the symbol $g_r(x)$. As the collection,

$$M = \{g_{1-\varepsilon}(x) : x \varepsilon U\}, \tag{112}$$

with ε sufficiently small is a covering and a partition in $INF(U)$, and

$$g_{1-\varepsilon}(x) =_{INF} Inf(x), \tag{113}$$

it follows that $\overline{x}^{\mu}_{M} = x = \underline{x}^{\mu}_{M}$. Applying the operator [.], we reach the conclusion that

$$\underline{x}_{M,\mu} = \underline{A}x. \tag{114}$$

Similarly, for the upper approximation,

$$\overline{x}^{M,\mu} = \overline{A}x. \tag{115}$$

For the argument, it suffices to notice that $[Inf(x)] = [x]_{IND(A)}$.

6.3 Fuzzy Partitions and Equivalences from Rough Inclusions

Because $y\mu_1 x$ is equivalent to $y\ el\ x$, we may interpret $y\mu_r x$ as a statement of fuzzy membership; we will write $\mu_x(y) = r$ to stress this interpretation.

Clearly, by its definition, a rough inclusion is a relation, or, it may be regarded as a generalized fuzzy membership function, that takes intervals of the form $[0, w]$ as its values, leading to a higher–level fuzzy set. That follows from (25). Hence, inequalities like $a \geq \tau(x, y)$ are interpreted as follows: the value of a belongs in the interval - the value of $\tau(x, y)$. We should also bear in mind that rough inclusions reflect information content of the underlying information system.

Thus, we may say, that rough inclusions induce globally a family of fuzzy sets $\{x : xel_{INF}INF(U)\}$ with fuzzy membership functions $\{\mu_x : xel_{INF}INF(U)\}$, in the above sense.

We assume additionally for the considered rough inclusion μ the f-transitivity property (47) with a t–norm f. Let us consider a relation τ on U defined as follows,

$$x\tau_r y \Leftrightarrow x\mu_r y \wedge y\mu_r x, \tag{116}$$

hence, τ_r is for each r a *tolerance relation.* The following properties hold,

$$x\tau_1 x, \tag{117}$$

$$x\tau_r y \Leftrightarrow y\tau_r x, \tag{118}$$

$$x\tau_r y \wedge y\tau_s z \Rightarrow x\tau_{f(r,s)} z. \tag{119}$$

We will write $\tau(x, y)$ instead of $\chi_{x,\tau}(y)$ in cases when we treat τ as a fuzzy set, except for cases when the latter notation is necessary.

We may paraphrase (117)–(119) in terms of the new notation,

$$\tau(x, x) = 1, \tag{120}$$

$$\tau(x, y) = \tau(y, x), \tag{121}$$

$$\tau(x, z) \geq f(\tau(x, y), \tau(y, z)). \tag{122}$$

Thus, τ is by (120), (121), and (122), an *f-fuzzy similarity*, see [27].

Following [27], we may define *similarity classes* $[x]_\tau$ as fuzzy sets satisfying,

$$\chi_{[x]_\tau}(y) = \tau(x, y). \tag{123}$$

The following are true,

$$\chi_{[x]_\tau}(x) = 1, \tag{124}$$

$$f(\chi_{[x]_\tau}(y), \tau(y, z)) \leq \chi_{[x]_\tau}(z), \tag{125}$$

$$f(\chi_{[x]_\tau}(y), \chi_{[x]_\tau}(z)) \leq \tau(y, z). \tag{126}$$

Equations (124, 125, 126) follow by corresponding properties of τ given by (117), (118), and (119).

Finally, the family $\{[x]_\tau : x \subseteq U\}$ does satisfy the requirements to be a *fuzzy partition* [5], [7], [23], [27], viz.,

$$\forall x \exists y . \chi_{[x]_\tau}(y) = 1, \tag{127}$$

$$[x]_\tau \neq [z]_\tau \Rightarrow \; max_y\{min\{\chi_{[x]_\tau}(y), \chi_{[z]_\tau}(y)\}\} < 1, \tag{128}$$

$$\bigcup_x [x]_\tau \times_f [x]_\tau = \tau, \tag{129}$$

where $A \times_f B$ denotes the fuzzy set defined via

$$\chi_{A \times_f B}(u, v) = f(\chi_A(u), \chi_B(v)) \tag{130}$$

and $\bigcup_x$ denotes the supremum over all values of x.

Indeed, (127), (128), (129) follow directly from properties of τ. For instance, (128) is justified as follows: if there was y with $\tau(x, y) = 1 = \tau(z, y)$, we would have $\tau(x, z), \tau(z, x) \geq f(1, 1) = 1$ hence $x =_U z$.

For (129), on one hand, given x, y, z, we have,

$$f(\tau(x, y), \tau(x, z)) = f(\tau(y, x), \tau(x, z)) \leq \tau(y, z)$$

by (122), hence, $\bigcup_x [x]_\tau \times_f [x]_\tau(y, z) \leq \tau(y, z)$.

On the other hand, letting $x = y$, we have,

$$[x]_\tau \times_f [x]_\tau(y, z) = [y]_\tau \times_f [y]_\tau(y, z) = f(\tau(y, y), \tau(y, z)) = f(1, \tau(y, z) = \tau(y, z)$$

by (120) and the property of every t–norm T that $T(1, u) = u$. Hence, $\bigcup_x [x]_\tau \times_f [x]_\tau(y, z) \leq \tau(y, z)$, which gives (129).

7 Conclusion

We may conclude our discussion; we have shown that rough mereological inclusions in the universe of an information system induce rough set approximations as well as fuzzy equivalence relations and partitions thus creating both rough as well as fuzzy framework. This concerns in particular the Menger as well as the Łukasiewicz rough inclusions, our exemplary rough inclusions.

References

1. C. Bishop.*Neural Networks for Pattern Recognition.*Clarendon, Oxford, repr. 1997.
2. G. Cattaneo.Generalized rough sets.Preclusivity fuzzy–intuitionistic (BZ) lattices. *Studia Logica*, **58**, 1997, pp. 47–77.
3. D. Ciucci. *Algebraic Structures for Rough Sets and Many–Valued Logics.* Manuscript. Univ. di Milano-Bicocca, 2003.
4. B. L. Clarke. A calculus of individuals based on “Connection”. *Notre Dame J. Formal Logic*, **22**, 1981, pp. 204–218.

5. D. Dubois and H. Prade. Putting rough sets and fuzzy sets together, in: R. Słowiński Ed. *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer, 1992, pp. 203–232.
6. S. Greco, B. Matarazzo, R. Słowiński, and J. Stefanowski. An algorithm for induction of decision rules consistent with the dominance principle. *Lecture Notes in Artificial Intelligence* **2005**, Springer Verlag, Berlin, 2001, pp. 304–313.
7. U. Hoehle. Quotients with respect to similarity relations. *Fuzzy Sets and Systems* **27**, 1988, pp. 31–44.
8. R. Intan, Y. Y. Yao and M. Mukaidono. Generalization of rough sets using weak fuzzy similarity relations. In:[9], pp. 37–46.
9. M. Inuiguchi, S. Hirano and S. Tsumoto Eds. *Rough Set Theory and Granular Computing*, Physica, Heidelberg, 2003.
10. M. Inuiguchi and T. Tanino. Two directions toward generalization of rough sets. In: [9], pp. 47–58.
11. S. Leśniewski. On the foundations of mathematics. *Topoi* **2**, 1982, pp. 7–52.
12. T.Y. Lin. Granular computing: Fuzzy logic and rough sets. In: [29], pp. 183–200.
13. C.-H. Ling. Representation of associative functions. *Publ. Math. Debrecen* **12**, 1965, pp. 189–212.
14. E. Marczewski. A general scheme of independence in mathematics. *Bull. Polish Acad. Math. Sci.* **6**, 1958, pp. 731–736.
15. S. K. Pal, L. Polkowski, and A. Skowron Eds. *Rough–Neurocomputing. Techniques for Computing with Words.* Springer Verlag, in print.
16. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer, 1992.
17. Z. Pawlak. Rough sets. *Intern. J. Comp. Inf. Sci.* **11**, 1982, pp. 341–356.
18. M, Novotný and Z. Pawlak. On rough equalities. *Bull. Polish Acad. Sci. Math.* **33**, 1985, pp. 99–104.
19. Z. Pawlak and A. Skowron. Rough membership functions, in: R. R. Yager, M. Fedrizzi, and J. Kacprzyk Eds. *Advances in the Dempster-Schafer Theory of Evidence.* Wiley, 1994, pp. 251–271.
20. L. Polkowski. A rough–neural computation model based on rough mereology. In: [15], pp. 85–108.
21. L. Polkowski. *Rough Sets. Mathematical Foundations.* Physica, Heidelberg, 2002.
22. L. Polkowski, A. Skowron. Rough mereology: a new paradigm for approximate reasoning. *International Journal of Approximate Reasoning* **15: 4**, 1997, pp. 333–365.
23. L. Valverde. On the structure of F–indistinguishability operators. *Fuzzy Sets and Systems* **17**, 1985, pp. 313–328.
24. A. N. Whitehead. *An Enquiry Concerning the Principles of Natural Knowledge.* Cambridge, 1919.
25. Y. Y. Yao. Generalized rough set models. In: L. Polkowski and A. Skowron Eds. *Rough Sets in Knowledge Discovery 1*, Physica, Heidelberg, 1998, pp. 286–320.
26. L. A. Zadeh. Fuzzy sets. *Information and Control* bf 8, 1965, pp. 338–353.
27. L. A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences* **3**, 1971, pp. 177–200.
28. L. A. Zadeh. Fuzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager Eds. *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 1979, pp. 3–18.
29. L. A. Zadeh and J. Kacprzyk Eds. *Computing with Words in Information/Intelligent Systems* **1**, Physica, Heidelberg, 1999.

Fuzzy Rough Sets Based on Residuated Lattices

Anna Maria Radzikowska[1] and Etienne E. Kerre[2]

[1] Faculty of Mathematics and Information Science
Warsaw university of Technology
Plac Politechniki 1, 00-661 Warsaw, Poland
annrad@mini.pw.edu.pl
[2] Dept. of Mathematics and Computer Science
Ghent University, Krijgslaan 281 (S9), B-9000 Gent, Belgium
Etienne.Kerre@rug.ac.be

Abstract. Rough sets were developed by Pawlak as a formal tool for representing and processing information in data tables. Fuzzy generalizations of rough sets were introduced by Dubois and Prade. In this paper, we consider *L–fuzzy rough sets* as a further generalization of the notion of rough sets. Specifically, we take a residuated lattice L as a basic structure. L–fuzzy rough sets are defined using the product operator and its residuum provided by the residuated lattice L. Depending on classes of binary fuzzy relations, we define several classes of L–fuzzy rough sets and investigate properties of these classes.

Keywords: L–fuzzy sets, rough sets, fuzzy relations, fuzzy logical operators, residuated lattices, modal operators, information operators.

1 Introduction

In real–life problems we are usually faced with situation where the available information is incomplete. Generally speaking, there are two inherent aspects of incompleteness of data: partiality and imprecision. On one side, not all relevant information is known, on the other one, user's data may be imprecise (e.g., expressed by means of linguistic terms, like "*quite good*", "*rather small*").

Rough sets theory was developed by Pawlak [11, 12] as a formal tool for representing and processing information in data tables – consequently, this formalism acts on partial information. The fuzzy set theory [23], on the other hand, offers a wide variety of techniques for analyzing imprecise data. It seems therefore natural to combine methods developed within both theories in order to construct hybrid structures capable to deal with both aspects of incompleteness. Such structures, called *fuzzy rough sets* and *rough fuzzy sets*, have been proposed in the literature [2, 3, 7, 19, 20].

In [14, 15] and [16] we have extensively investigated fuzzy rough sets taking the interval [0,1] as a basic structure. However, as Goguen [5] pointed out, in some situations it may be impossible to use the linearly ordered set [0,1] to represent degrees of membership. From this reason, the concept of an *L–fuzzy set* was then introduced.

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 278–296, 2004.

In this paper, we consider fuzzy rough sets defined on the basis of *residuated lattices.* The importance of these algebras results from the role they play in fuzzy logics [6, 22]. Due to tight relations between rough set theory and semantics for modal logics, this approach allows for constructing and investigating various systems of fuzzy modal logics.

The paper is organized as follows. In Sect. 2 we recall basic properties of residuated lattices and triangular norms. In Sect. 3 we briefly recall foundations of L–fuzzy sets and L–fuzzy relations. Fuzzy rough sets are defined in Sect. 4, and their basic properties are given in Sect. 5. In the next section we discuss several classes of fuzzy rough sets. The chapter is completed by concluding remarks.

2 Residuated Lattices

Let $(L, \leqslant)$ be a poset and let $\otimes$ be a binary operation in L. The following two binary operations $\rightarrow$ and $\Rightarrow$ in L, called the *left* and *right residuum of* $\otimes$, respectively, are such that for all $x, y \in L$,

(Rc.1) $x \otimes y \leqslant z$ iff $x \leqslant y \rightarrow z$
(Rc.2) $x \otimes y \leqslant z$ iff $y \leqslant x \Rightarrow z$.

The conditions **(Rc.1)** and **(Rc.2)** are called the *residuation conditions.*
It is easy to show (see [22]) that if the respective residua of $\otimes$ exist then

$$x \rightarrow y = \sup\{z \in L : x \otimes z \leqslant y\}$$
$$x \Rightarrow y = \sup\{z \in L : z \otimes z \leqslant y\}.$$

Clearly, if $\otimes$ is commutative then $\rightarrow = \Rightarrow$.

Definition 1. *A* monoid *is a structure* $(M, \otimes, \mu)$*, where* M *is a non–empty universe,* $\otimes : M \times M \rightarrow M$ *is an associative operation and* μ *is a neutral element of* $\otimes$*, i.e. for every* $x \in M$*,* $\mu \otimes x = x \otimes \mu = x$*.* □

A monoid $(M, \otimes, \mu)$ is called *commutative* iff $\otimes$ is commutative.

Definition 2. *A* residuated lattice *is a structure* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *such that*

- $(L, \otimes, \top)$ *is a monoid*
- $(L, \wedge, \vee, 0, 1)$ *is a bounded lattice with the greatest element* 1 *and the smallest element* 0
- $\rightarrow$ *and* $\Rightarrow$ *are the left and the right residuum of* $\otimes$*, respectively.* □

A residuated lattice $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ is called

- *commutative* iff $\otimes$ is commutative
- *integral* iff $\top = 1$
- *complete* iff the underlying lattice $(L, \wedge, \vee, 1, 0)$ is complete.

Given a lattice $(L, \wedge, \vee, 0, 1)$ and its ordering $\leqslant$, for any $x, y \in L$ we will write $x < y$ (resp. $x > y$) iff $x \neq y$ and $x \leqslant y$ (resp. $y \leqslant x$). If L is commutative then it will be just written $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1, \top)$, if it is also integral, we will write $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$.

The following proposition provides basic properties of residuated lattices [6, 22].

Lemma 1. *In every residuated lattice* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *the following conditions hold for every* $x, y, z \in \mathfrak{X}$*:*

(i) $\otimes$ *is isotone in both arguments*
(ii) $\rightarrow$ *is antitone in the 1st argument and isotone in the 2nd argument*
(iii) $0 \otimes x = 0$
(iv) $(x \otimes y) \leqslant (x \wedge y)$
(v) $(x \rightarrow (x \otimes y)) \geqslant y$
(vi) $(x \otimes (x \rightarrow y)) \leqslant y$. ■

Lemma 2. *In every integral residuated lattice* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1)$ *the following conditions hold for every* $x, y, z \in \mathfrak{X}$*:*

(i) $(1 \rightarrow x) = x$, $(x \rightarrow x) = 1$, $(0 \rightarrow 1) = 1$, $(0 \rightarrow 0) = 1$
(ii) $(x \rightarrow y) = 1$ *iff* $x \leqslant y$
(iii) *if* L *is commutative then* $(x \rightarrow (y \rightarrow z)) = ((x \otimes y) \rightarrow z)$. ■

Given a residuated lattice $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$, let us define two unary operators $\neg_l$ and $\neg_r$, referred to as the *precomplement operators*, by: for every $x \in L$,

$$\neg_l x = x \rightarrow 0$$
$$\neg_r = x \Rightarrow 0$$

Clearly, if L is commutative then $\neg_l = \neg_r$.

Precomplement operations are generalizations of the pseudo–complement in a lattice (see [17]). If $\wedge = \otimes$ then $\rightarrow$ is the relative pseudo–complement, $\neg = \neg_l = \neg_r$ is the pseudo–complement and $(W, \wedge, \vee, \rightarrow, \neg, 0, 1)$ is a Heyting algebra.
The following lemma provides basic properties of the precomplement operator.

Lemma 3. *In every commutative and integral residuated lattice* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *the following conditions hold for all* $x, y, z \in L$*:*

(i) $\neg\neg x \geqslant x$
(ii) $(x \rightarrow y) \leqslant (\neg y \rightarrow \neg x)$
(iii) $((x \rightarrow y) \otimes (y \rightarrow z)) \leqslant (x \rightarrow z)$
(iv) $(x \rightarrow \neg y) = \neg(x \otimes y)$. ■

Lemma 4. *In every complete, commutative and integral residuated lattice* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *the following conditions hold for every* $z \in L$ *and for every indexed family* $(x_i)_{i \in I}$ *of elements of* L*:*

(i) $(\sup_{i \in I} x_i \rightarrow z) = \inf_{i \in I}(x_i \rightarrow z)$
(ii) $z \rightarrow (\inf_{i \in I} x_i) = \inf_{i \in I}(z \rightarrow x_i)$

- **(iii)** $\neg \sup_{i\in I} x_i = \inf_{i\in I} \neg x_i$
- **(iv)** $\neg \inf_{i\in I} x_i \geqslant \sup_{i\in I} \neg x_i$
- **(v)** $z \otimes (\sup_{i\in I} x_i) = \sup_{i\in I}(z \otimes x_i)$
- **(vi)** $z \otimes (\inf_{i\in I} x_i) \leqslant \inf_{i\in I}(z \otimes x_i)$
- **(vii)** $(\inf_{i\in I} x_i) \rightarrow (\sup_{i\in I} y_i) \geqslant \sup_{i\in I}(x_i \rightarrow y_i)$. ■

Residuated lattices are the weak structures that provide an algebraic framework for degrees of membership to fuzzy sets and, in a more general setting, degrees of certainty of assertions in fuzzy theories. The other classes of algebras, analogous to the algebras studied in [4] and [6], are obtained from residuated lattices by postulating some additional axioms.

Definition 3. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a residuated lattice. We say that* L *is an* MTL–algebra *iff the following condition holds:*

$$(x \rightarrow y) \vee (y \rightarrow x) = 1 \;\; \textit{for all}\, x, y \in L. \quad \square \tag{1}$$

Condition (1) is called a *prelinearity condition* [22].

The acronym *MTL* means *Monoidal t–norm Logic*. Monoidal t–norm logic MTL has been introduced in [4].

The following lemma provides some properties of MTL–algebras.

Lemma 5. *In any MTL–algebra* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *the following conditions hold for all* $x, y, z \in L$

- **(i)** $x \otimes y \leqslant (x \otimes x) \vee (y \otimes y)$
- **(ii)** $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
- **(iii)** $x \rightarrow (y \vee z) = (x \rightarrow y) \vee (x \rightarrow z)$
- **(iv)** $(x \wedge y) \rightarrow z = (x \rightarrow z) \vee (y \rightarrow z)$
- **(v)** *if* L *is complete then for any indexed family* $(z_i)_{i\in I}$ *of elements of* L, $\sup_{i\in I}(z_i \otimes z_i) = (\sup_{i\in I} z_i) \otimes (\sup_{i\in I} z_i)$. ■

2.1 Triangular Norms

A typical example of monoid operators that play an important role in fuzzy sets theory are triangular norms.

Definition 4. *A* t*riangular norm (t–norm, for short) is a mapping* $\star : [0,1]^2 \rightarrow [0,1]$ *satisfying the following conditions:*

- **(T.1)** $\star$ *is commutative and associative*
- **(T.2)** $\star$ *is isotone in both arguments, that is for* $x \leqslant y$ *it holds* $x \star z \leqslant y \star z$, *where* $\leqslant$ *is the natural ordering on reals*
- **(T.3)** $1 \star x = x$ *for every* $x \in [0,1]$. □

Well–known t–norms are:

- **(i)** the standard *min* operation $x \star_Z y = \min\{x, y\}$,
- **(ii)** the algebraic product $x \star_P y = x \cdot y$

(iii) the Łukasiewicz t–norm $x \star_L y = \max\{0, x+y-1\}$
(iv) the drastic t–norm

$$x \star_D y = \begin{cases} x \text{ if } y = 1 \\ y \text{ if } x = 1 \\ 0 \text{ otherwise.} \end{cases}$$

It can be shown that $\star_Z$ is the largest t–norm, whereas the $\star_D$ is the smallest t–norm.

A t–norm is called *left–continuous* iff it has left–continuous partial mappings.

Proposition 1. [4] *Let* $\mathcal{L} = ([0,1], max, min, \star, \rightarrow, 0, 1)$ *be a structure such that* $\star$ *is a t–norm and* $\rightarrow$ *is its residuum. Then*

1. *If* $\star$ *is left–continuous then* $\mathcal{L}$ *is a residuated lattice.*
2. *In* $\mathcal{L}$ *the following identities hold:*

$$\max\{x \rightarrow y\,,\; y \rightarrow x\} = 1$$
$$\max\{x, y\} = \min\{((x \rightarrow y) \rightarrow y)\,,\; ((y \rightarrow x) \rightarrow x).$$

3. *If* $\star$ *is continuous then* $x \star (x \rightarrow y) = \min\{x, y\}$ *holds in* $\mathcal{L}$. ■

For a left–continuous t–norm $\star$, its residuum is referred to as *residual implication.* Three well–known residual implication, being the residua of $\star_Z$, $\star_P$ and $\star_L$, respectively, are:

(i) the Gödel implication $x \rightarrow_Z y = \begin{cases} 1 \text{ if } x \leqslant y \\ y \text{ otherwise} \end{cases}$
(ii) the Gaines implication $x \rightarrow_P y = \begin{cases} 1 \text{ if } x \leqslant y \\ \frac{y}{x} \text{ otherwise} \end{cases}$
(iii) the Łukasiewicz implication $x \rightarrow_L y = \min\{1, 1-x+y\}$.

3 Fuzzy Sets and Fuzzy Relations

Let L be a lattice. *L–fuzzy sets* was introduced by Goguen [5] as a generaliaation of the notion of Zadeh's fuzzy sets [23]. Below we provide more specific definition of this notion, restricting ourselves to residuated lattices taken as a basic structure.

Definition 5. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a residuated lattice and let* $\mathfrak{X}$ *be a nonempty universe. A mapping* $A : \mathfrak{X} \rightarrow L$ *is called an* L*–fuzzy set in* $\mathfrak{X}$. □

The family of all L–fuzzy sets in $\mathfrak{X}$ will be denoted by $\mathcal{F}_L(\mathfrak{X})$.

The following are the most popular operations on L–fuzzy sets: for all $A, B \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$,

$$(A \sqcap_L B)(x) = A(x) \otimes B(x)$$
$$(A \cap_L B)(x) = A(x) \wedge B(x)$$
$$(A \cup_L B)(x) = A(x) \vee B(x)$$
$$(co_L A)(x) = \neg A(x).$$

For any $A, B \in \mathcal{F}_L(\mathfrak{X})$, we will write $A \subseteq_L B$ to denote $A(x) \leqslant B(x)$ for every $x \in \mathfrak{X}$.

Let $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ be a complete residuated lattice. For any family $(A_i)_{i\in I}$ of L–fuzzy sets, we will write $\bigcup_{i\in I} A_i$ and $\bigcap_{i\in I} A_i$ to denote the L–fuzzy sets in $\mathfrak{X}$ given by

$$(\textstyle\bigcup_{i\in I} A_i)(x) = \sup_{i\in I} A(x)$$
$$(\textstyle\bigcap_{i\in I} A_i)(x) = \inf_{i\in I} A(x).$$

3.1 Fuzzy Relations

Definition 6. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a residuated lattice and let* $\mathfrak{X}$ *be a non–empty universe. Any mapping* $R : \mathfrak{X}^n \rightarrow L$ *for* $n \geqslant 2$, *is called an f* ***L***–fuzzy relation. *For* $n = 2$ *R is called a* binary ***L***–fuzzy relation. □

The family of all binary L–fuzzy relations on $\mathfrak{X}$will be denoted by $\mathcal{R}_L(\mathfrak{X})$.
Let $R \in \mathcal{R}_L(\mathfrak{X})$. We say that R is

- *serial* iff $\sup_{y\in\mathfrak{X}} R(x, y) = 1$ for every $x \in \mathfrak{X}$ [1]
- *reflexive* iff $R(x, x) = 1$ for every $x \in \mathfrak{X}$
- *symmetric* iff $R(x, y) = R(y, x)$ for all $x, y \in \mathfrak{X}$
- *Euclidean* iff $R(z, x) \otimes R(z, y) \leqslant R(x, y)$ for all $x, y, z \in \mathfrak{X}$
- *transitive* iff $R(x, z) \otimes R(z, y) \leqslant R(x, y)$ for all $x, y, z \in \mathfrak{X}$.

A binary L–fuzzy relation R is called an *L–tolerance* relation iff R is reflexive and symmetric; if an L–tolerance relation is also transitive then it is called an *L–equivalence* relation.

Given $R \in \mathcal{R}_L(\mathfrak{X})$ and $x \in \mathfrak{X}$, we will write xR to denote the L–fuzzy set in $\mathfrak{X}$ such that $(xR)(y) = R(x, y)$ for every $y \in \mathfrak{X}$.

4 Fuzzy Rough Sets

Definition 7. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a complete residuated lattice,* $\mathfrak{X}$ *be a nonempty universe and let* $R \in \mathcal{R}_L(\mathfrak{X})$. *A pair* Fas $= (\mathfrak{X}, R)$ *is called an* ***L***–fuzzy approximation space. □

By a *fuzzy approximation space* we mean an L–fuzzy approximation space for some residuated lattice L.

Depending on a class of fuzzy relations, we have the corresponding classes of L–fuzzy approximation spaces. Namely, we will consider the following fuzzy approximation spaces

- Lfas: with serial L–fuzzy relations
- Rfas: with L–fuzzy reflexive relations
- Sfas: with symmetric L–fuzzy relations

[1] In this case the lattice L is assumed to be complete.

– EFAS: with Euclidean L–fuzzy relations
– TFAS: with transitive L–fuzzy relations
– QFAS: with equivalence L–fuzzy relations.

Definition 8. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a complete residuated lattice and let* $\mathrm{FAS} = (\mathfrak{X}, R)$ *be an L–fuzzy approximation space. Define the following two mappings* $\underline{\mathrm{FAS}}_L, \overline{\mathrm{FAS}}^L : \mathcal{F}_L(\mathfrak{X}) \rightarrow \mathcal{F}_L(\mathfrak{X})$*, called a* lower *and* upper L–fuzzy rough approximators*, respectively, as follows: for every* $A \in \mathcal{F}_L(\mathfrak{X})$ *and every* $x \in \mathfrak{X}$*,*

$$\begin{aligned} \underline{\mathrm{FAS}}_L(A)(x) &= \inf_{y \in \mathfrak{X}}(R(x,y) \rightarrow A(y)) \\ \overline{\mathrm{FAS}}^L(A)(x) &= \sup_{y \in \mathfrak{X}}(R(x,y) \otimes A(y)). \end{aligned}$$

$\underline{\mathrm{FAS}}_L(A)$ (*resp.* $\overline{\mathrm{FAS}}^L(A)$) *is called an* lower (*resp.* upper) L–fuzzy rough approximation of A. □

Notice that the lower and the upper L–fuzzy rough approximator coincide with fuzzy box and fuzzy diamond modal operators, respectively. Consequently, these operations correspond to the notions of *necessity* and *possibility*, respectively. Then, for any $A \in \mathcal{F}_L(\mathfrak{X})$ and for any $x \in \mathfrak{X}$, $\underline{\mathrm{FAS}}_L(A)(x)$ (resp. $\overline{\mathrm{FAS}}^L(A)(x)$) reflects the degree to which x *certainly* (resp. *possibly*) belongs to A,

Definition 9. *Let L be a complete residuated lattice and* $\mathrm{FAS} = (\mathfrak{X}, R)$ *an L–fuzzy approximation space. A pair* $(L, U) \in \mathcal{F}_L(\mathfrak{X}) \times \mathcal{F}_L(\mathfrak{X})$ *such that* $L = \underline{\mathrm{FAS}}_L(A)$ *and* $U = \overline{\mathrm{FAS}}^L(A)$ *for some* $A \in \mathcal{F}_L(\mathfrak{X})$*, is called an* L–fuzzy rough set in FAS. □

We say that a pair (L, U) is a *fuzzy rough set* iff it is an L–fuzzy rough set for some complete residuated lattice L.

Let $\mathcal{K}$ be a class of L–fuzzy relations. We will write $FRS(\mathcal{K})$ to denote the class of all L–fuzzy rough sets in $\mathcal{K}$–fuzzy approximation spaces.

5 Basic Properties of Fuzzy Rough Sets

Proposition 2. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \Rightarrow, 0, 1, \top)$ *be a complete residuated lattice. For every L–fuzzy approximation space* $\mathrm{FAS} = (\mathfrak{X}, R)$*, every* $A \in \mathcal{F}_L(\mathfrak{X})$ *and for every indexed family* $(A_i)_{i \in I}$ *the following conditions hold:*

(i) $\overline{\mathrm{FAS}}^L(\emptyset) = \emptyset, \quad \underline{\mathrm{FAS}}_L(\mathfrak{X}) = \mathfrak{X}$

(ii) *if* $A \subseteq_L B$ *then* $\underline{\mathrm{FAS}}_L(A) \subseteq_L \underline{\mathrm{FAS}}_L(B)$ *and* $\overline{\mathrm{FAS}}^L(A) \subseteq_L \underline{\mathrm{FAS}}_L(B)$

(iii) $\underline{\mathrm{FAS}}_L(A) \subseteq_L co_L \overline{\mathrm{FAS}}^L(co_L A)$
$\overline{\mathrm{FAS}}^L(A) \subseteq_L co_L \underline{\mathrm{FAS}}_L(co_L A)$
$co_L \overline{\mathrm{FAS}}^L(A) = \underline{\mathrm{FAS}}_L(co_L A)$

(iv) $\overline{\mathrm{FAS}}^L(\bigcup_{i \in I} A_i) = \bigcup_{i \in I} \overline{\mathrm{FAS}}^L(A_i)$
$\underline{\mathrm{FAS}}_L(\bigcap_{i \in I} A_i) = \bigcap_{i \in I} \underline{\mathrm{FAS}}_L(A)_i)$.

Proof.

(i) Results from Lemma 1**(iii)** and 2**(i)**.

(ii) Follows from Lemma 1**(i)** and 1**(ii)**.

(iii) For every $x \in \mathfrak{X}$ we have

$$\begin{aligned}(co_L \underline{\text{FAS}}_L(co_L A))(x) &= \neg(\inf_{y\in\mathfrak{X}}(R(x,y) \to \neg A(y))) & \\ &= \neg \inf_{y\in\mathfrak{X}} \neg(R(x,y) \otimes A(y)) & \text{Lemma } 3\textbf{(iv)} \\ &\geqslant \sup_{y\in\mathfrak{X}} \neg\neg(R(x,y) \otimes A(y)) & \text{Lemma } 4\textbf{(iv)} \\ &\geqslant \sup_{y\in\mathfrak{X}}(R(x,y) \otimes A(y)) & \text{Lemma } 3\textbf{(i)}\end{aligned}$$

so $\overline{\text{FAS}}^L(A) \subseteq_L co_L \underline{\text{FAS}}_L(co_L A)$.
The second inclusion can be proved in the similar way.

Also, for every $x \in \mathfrak{X}$,

$$\begin{aligned}(co_L \overline{\text{FAS}}^L(A))(x) &= \neg \sup_{y\in\mathfrak{X}}(R(x,y) \otimes A(y)) & \\ &= \inf_{y\in\mathfrak{X}} \neg(R(x,y) \otimes A(y)) & \text{Lemma } 4\textbf{(iii)} \\ &= \inf_{y\in\mathfrak{X}}(R(x,y) \to \neg A(y)) & \text{Lemma } 3\textbf{(iv)} \\ &= \underline{\text{FAS}}_L(co_L \text{A})(\text{x}). &\end{aligned}$$

Hence $co_L \overline{\text{FAS}}^L(A) = \underline{\text{FAS}}_L(co_L A)$.

(iv) For every family $(A_i)_{i\in I}$ of L–fuzzy sets in $\mathfrak{X}$ and every $x \in \mathfrak{X}$ we have:

$$\begin{aligned}(\underline{\text{FAS}}_L(\textstyle\bigcap_{i\in I} A_i))(x) &= \inf_{y\in\mathfrak{X}}(R(x,y) \to \inf_{i\in I} A_i(y)) \\ &= \inf_{i\in I} \inf_{y\in\mathfrak{X}}(R(x,y) \to A_i(y)) \\ &= \inf_{i\in I} \underline{\text{FAS}}_L(A_i)(x) \\ &= (\textstyle\bigcap_{i\in I} \underline{\text{FAS}}_L(A_i))(x)\end{aligned}$$

by Lemma 4**(ii)**. Also, for every $(A_i)_{i\in I}$,

$$\begin{aligned}\overline{\text{FAS}}^L(\textstyle\bigcup_{i\in I} A_i)(x) &= \sup_{y\in\mathfrak{X}}(R(x,y) \otimes \sup_{i\in I} A_i(y)) \\ &= \sup_{i\in I} \sup_{y\in\mathfrak{X}}(R(x,y) \otimes A_i(y)) \\ &= \sup_{i\in I} \overline{\text{FAS}}^L(A_i)(x) \\ &= (\textstyle\bigcup_{i\in I} \overline{\text{FAS}}^L(A_i))(x)\end{aligned}$$

by Lemma 4**(v)**. ■

6 Selected Classes of Fuzzy Rough Sets

In this section we consider several classes of L–fuzzy rough sets. More specifically, we focus on particular classes of fuzzy approximation spaces and investigate L–fuzzy rough sets in these spaces.

6.1 Fuzzy Rough Sets in Lfas Spaces

Let us start with fuzzy approximation spaces which underlying relations are serial. For these spaces the following property holds.

Proposition 3. *For every complete, commutative and integral MTL–algebra* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *and for every L–fuzzy approximation space* Fas$=(\mathfrak{X}, R)$ *the following conditions are equivalent:*

(i) $\mathrm{Fas} \in \mathrm{Lfas}$

(ii) $\underline{\mathrm{Fas}}_L(A) \subseteq_L \overline{\mathrm{Fas}}^L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$.

Proof.

(i) ⇒ **(ii)** For every $A \in \mathcal{F}_L(\mathfrak{X})$ and every $x \in \mathfrak{X}$ we have:

$$\begin{aligned}&(\underline{\mathrm{Fas}}_L(A)(x) \rightarrow \overline{\mathrm{Fas}}^L(A)(x))\\ &\quad = (\inf_{y\in\mathfrak{X}}(R(x,y) \rightarrow A(y))) \rightarrow (\sup_{y\in\mathfrak{X}}(R(x,y) \otimes A(y)))\\ &\quad \geqslant \sup_{y\in\mathfrak{X}}[(R(x,y) \rightarrow A(y)) \rightarrow (R(x,y) \otimes A(y)]\end{aligned}$$

by Lemma 4**(vii)**. Next, by Lemma 2**(iii)**, $A(y) \geqslant R(x,y) \otimes (R(x,y) \rightarrow A(y))$, so $R(x,y) \otimes A(y) \geqslant R(x,y) \otimes R(x,y) \otimes (R(x,y) \rightarrow A(y))$. Then we get

$$\begin{aligned}&(\underline{\mathrm{Fas}}_L(A)(x) \rightarrow \overline{\mathrm{Fas}}^L(A)(x))\\ &\quad \geqslant \sup_{y\in\mathfrak{X}}[(R(x,y) \rightarrow A(y)) \rightarrow [(R(x,y) \otimes R(x,y)) \otimes (R(x,y) \rightarrow A(y))]]\\ &\quad \geqslant \sup_{y\in\mathfrak{X}}(R(x,y) \otimes R(x,y))\end{aligned}$$

by Lemma 1**(v)**. Since L is an MTL algebra, by Lemma 5**(v)** we have

$$\sup_{y\in\mathfrak{X}}(R(x,y) \otimes R(x,y)) = \sup_{y\in\mathfrak{X}}R(x,y) \otimes \sup_{y\in\mathfrak{X}}R(x,y) = 1,$$

since R is serial. So we get $\underline{\mathrm{Fas}}_L(A)(x) \rightarrow \overline{\mathrm{Fas}}^L(A)(x) = 1$ for every $x \in \mathfrak{X}$. By Lemma 2**(ii)** this means that for every $x \in \mathfrak{X}$, $\underline{\mathrm{Fas}}_L(A)(x) \leqslant \overline{\mathrm{Fas}}^L(A)(x)$, so $\underline{\mathrm{Fas}}_L(A) \subseteq_L \overline{\mathrm{Fas}}^L(A)$.

(ii) ⇒ **(i)** Assume that $\mathrm{Fas} = (\mathfrak{X}, R)$ is such that $\mathrm{Fas} \notin \mathrm{Lfas}$, that is R is not serial. This means that for some $x_0 \in \mathfrak{X}$, $\sup_{y\in\mathfrak{X}} R(x_0,y) \neq 1$. Take $A = x_0R$. We will show that $\underline{\mathrm{Fas}}_L(A) \not\subseteq_L \overline{\mathrm{Fas}}^L(A)$.

Note that

$$\begin{aligned}&\underline{\mathrm{Fas}}_L(A)(x_0) \rightarrow \overline{\mathrm{Fas}}^L(A)(x_0)\\ &\quad = (\inf_{y\in\mathfrak{X}}(R(x_0,y) \rightarrow R(x_0,y))) \rightarrow (\sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(x_0,y))).\end{aligned}$$

By Lemma 2**(ii)**, $R(x_0,y) \rightarrow R(x_0,y) = 1$, so $\underline{\mathrm{Fas}}_L(A)(x_0) = 1$. Hence we have:

$$\begin{aligned}\underline{\mathrm{Fas}}_L(A)(x_0) \rightarrow \overline{\mathrm{Fas}}^L(A)(x_0) &= 1 \rightarrow \sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(x_0,y))\\ &= \sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(x_0,y))\end{aligned}$$

by Lemma 2(**i**). Applying again Lemma 5(**v**) we get

$$\begin{aligned}\sup_{y\in\mathfrak{X}}(R(x_0,y)\otimes R(x_0,y)) &= (\sup_{y\in\mathfrak{X}}R(x_0,y))\otimes(\sup_{y\in\mathfrak{X}}R(x_0,y))\\ &\leqslant \sup_{y\in\mathfrak{X}}R(x_0,y)<1.\end{aligned}$$

Then $\underline{\mathrm{FAS}}_L(A)(x_0)\rightarrow\overline{\mathrm{FAS}}^L(A)(x_0)<1$, which by Lemma 2(**ii**) means that $\underline{\mathrm{FAS}}_L(A)(x_0)\nleq\overline{\mathrm{FAS}}^L(A)(x_0)$, which implies the result. ■

Proposition 3 implies the following

Corollary 1. *Let* $(L,\wedge,\vee,\otimes,\rightarrow,0,1)$ *be a complete, commutative and integral MTL–algebra and let* (A,B) *be an* L*–fuzzy rough set in* $\mathrm{FAS}\in\mathrm{LFAS}$. *Then* $A\subseteq_L B$. ■

6.2 Fuzzy Rough Approximations in RFAS Spaces

Now let us consider L–fuzzy rough sets in approximation spaces, where the underlying fuzzy relation is reflexive.

Proposition 4. *For any complete and integral residuated lattice* $(L,\wedge,\vee,\otimes,\rightarrow,\Rightarrow,0,1)$ *and for any* L*–fuzzy approximation space* $\mathrm{FAS}=(\mathfrak{X},R)$ *such that* $\mathrm{FAS}\in\mathrm{RFAS}$*:*

(i) $\underline{\mathrm{FAS}}_L(\emptyset)=\emptyset$

(ii) $\overline{\mathrm{FAS}}^L(\mathfrak{X})=\mathfrak{X}$.

Proof.

(i) For every $x\in\mathfrak{X}$,

$$\underline{\mathrm{FAS}}_L(\emptyset)(x)=\inf_{y\in\mathfrak{X}}(R(x,y)\rightarrow 0)\leqslant R(x,x)\rightarrow 0=1\rightarrow 0=0$$

by Lemma 2(**i**).

(ii) For every $x\in\mathfrak{X}$,

$$\overline{\mathrm{FAS}}^L(\mathfrak{X})(x)=\sup_{y\in\mathfrak{X}}(R(x,y)\otimes 1)\geqslant R(x,x)\otimes 1=R(x,x)=1,$$

since, by assumption, R is reflexive. ■

Proposition 5. *For any complete and integral residuated lattice* $(L,\wedge,\vee,\otimes,\rightarrow,\Rightarrow,0,1)$ *and for any* L*–fuzzy approximation space* $\mathrm{FAS}=(\mathfrak{X},R)$ *the following conditions are equivalent:*

(i) $\mathrm{FAS}\in\mathrm{RFAS}$

(ii) $\underline{\mathrm{FAS}}_L(A)\subseteq_L A$ *for every* $A\in\mathcal{F}_L(\mathfrak{X})$

(iii) $A\subseteq_L\overline{\mathrm{FAS}}^L(A)$ *for every* $A\in\mathcal{F}_L(\mathfrak{X})$.

Proof.

(i)$\Rightarrow$(ii) For every $A\in\mathcal{F}_L(\mathfrak{X})$ and for every $x\in\mathfrak{X}$,

$$\underline{\mathrm{FAS}}_L(A)(x)=\inf_{y\in\mathfrak{X}}(R(x,y)\rightarrow A(y))\leqslant R(x,x)\rightarrow A(x)=1\rightarrow A(x)=A(x)$$

by assumption and Lemma 2(**i**). Hence $\underline{\mathrm{FAS}}_L(A)\subseteq_L A$.

(ii) ⇒ (i) Assume that $\textsc{Fas} \notin \textsc{Rfas}$, that is R is not reflexive. It means that $R(x_0, x_0) < 1$ for some $x_0 \in \mathfrak{X}$. Consider $A = x_0R$. Then we have

$$\underline{\textsc{Fas}}_L(A)(x_0) = \inf_{y\in\mathfrak{X}}(R(x_0,y) \to A(y)) = \inf_{y\in\mathfrak{X}}(R(x_0,y) \to R(x_0,y)) = 1$$

by Lemma 2(**ii**). However, $A(x_0) = R(x_0, x_0) < 1$, so $\underline{\textsc{Fas}}_L(A)(x_0) > A(x_0)$, which gives $\underline{\textsc{Fas}}_L(A) \not\subseteq_L A$.

(i) ⇒ (iii) For every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$ we have

$$\overline{\textsc{Fas}}^L(A)(x) = \sup_{y\in\mathfrak{X}}(R(x,y) \otimes A(y)) \geqslant R(x,x) \otimes A(x) = 1 \otimes A(x) = A(x),$$

so $A \subseteq_L \overline{\textsc{Fas}}^L(A)$.

(iii) ⇒ (i) Assume that $\textsc{Fas} \notin \textsc{Lfas}$, i.e. R is not reflexive. This means that $R(x_0, x_0) < 1$ for some $x_0 \in \mathfrak{X}$. For $A = \{x_0\}$ we have

$$\begin{aligned}\overline{\textsc{Fas}}^L(A)(x_0) &= \sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes A(y)) = R(x_0,x_0) \otimes 1\\ &= R(x_0,x_0) < 1 = A(x_0).\end{aligned}$$

Therefore $A \not\subseteq_L \overline{\textsc{Fas}}^L(A)$. ■

Note that for properties stated in the above propositions, commutativity of L is not necessary.

6.3 Fuzzy Rough Sets in Sfas Spaces

Proposition 6. *For any complete and integral residuated lattice* $(L, \wedge, \vee, \otimes, \to, \Rightarrow, 0, 1)$ *and for any L–fuzzy approximation space* $\textsc{Fas}=(\mathfrak{X}, R)$ *the following conditions are equivalent:*

(i) $\textsc{Fas} \in \textsc{Sfas}$

(iii) $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L A$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$

(ii) $A \subseteq_L \underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$.

Proof. **(i) ⇒ (ii)** For every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$,

$$\begin{aligned}&\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(x)\\ &\quad= \sup_{y\in\mathfrak{X}}(R(x,y) \otimes \inf_{z\in\mathfrak{X}}(R(y,z) \to A(z)))\\ &\quad\leqslant \sup_{y\in\mathfrak{X}}(R(x,y) \otimes (R(y,x) \to A(x))) && \text{Lemma 1}(\mathbf{i})\\ &\quad= \sup_{y\in\mathfrak{X}}(R(x,y) \otimes (R(x,y) \to A(x))) && \text{symmetry of } R\\ &\quad\leqslant A(x) && \text{Lemma 1}(\mathbf{vi}).\end{aligned}$$

Hence, indeed, $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L A$.

(ii) ⇒ (i) Assume that $\textsc{Fas} \notin \textsc{Sfas}$, that is there exist $x_0, y_0 \in \mathfrak{X}$ such that $R(x_0, y_0) \neq R(y_0, x_0)$. Consider the following two cases:

Case 1: Assume $R(x_0,y_0) \leqslant R(y_0,x_0)$. Since $R(x_0,y_0) \neq R(y_0,x_0)$, we have $R(y_0,x_0) < R(x_0,y_0)$, so by Lemma 2**(ii)**,

$$R(y_0,x_0) \to R(x_0,y_0) < 1. \tag{2}$$

For $A = x_0R$, by Lemma 1**(ii)**, we have:

$$\begin{aligned}
&\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(y_0) \to A(y_0) \\
&= [\sup_{x\in\mathfrak{X}}(R(y_0,x) \otimes \inf_{z\in\mathfrak{X}}(R(x,z) \to R(x_0,z)))] \to R(x_0,y_0) \\
&\leqslant (R(y_0,x_0) \otimes \inf_{z\in\mathfrak{X}}(R(x_0,z) \to R(x_0,z))) \to R(x_0,y_0) && \text{Lemma 1(ii)} \\
&= R(y_0,x_0) \otimes 1 \to R(x_0,y_0) && \text{Lemma 2(ii)} \\
&= R(y_0,x_0) \to R(x_0,y_0) && L \text{ is integral} \\
&< 1. && \text{from (2).}
\end{aligned}$$

Applying again Lemma 2**(ii)** we get $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(y_0) \nleqslant A(y_0)$, which gives $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \not\subseteq_L A$.

Case 2: Let $R(x_0,y_0) \nleqslant R(y_0,x_0)$. Then $R(x_0,y_0) \to R(y_0,x_0) < 1$ by Lemma 2**(ii)**. Take $A = y_0R$. Similarly we have:

$$\begin{aligned}
&\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(x_0) \to A(x_0) \\
&\qquad = (\sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes \inf_{z\in\mathfrak{X}}(R(y,z) \to R(y_0,z)))) \to R(y_0,x_0) \\
&\qquad \leqslant R(x_0,y_0) \to R(y_0,x_0) < 1,
\end{aligned}$$

so $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(x_0) \nleqslant A(x_0)$. Again $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \not\subseteq_L A$.

(i) $\Rightarrow$ **(iii)** For every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$,

$$\begin{aligned}
&\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))(x) \\
&\qquad = \sup_{y\in\mathfrak{X}}(R(x,y) \otimes \inf_{z\in\mathfrak{X}}(R(y,z) \to A(z))) \\
&\qquad \leqslant \sup_{y\in\mathfrak{X}}(R(x,y) \otimes (R(y,x) \to A(x))) && \text{Lemma 1(i)} \\
&\qquad = \sup_{y\in\mathfrak{X}}(R(x,y) \otimes (R(x,y) \to A(x))) && \text{symmetry of } R \\
&\qquad \leq 0.5ex]A(x) && \text{Lemma 1(vi).}
\end{aligned}$$

Hence $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L A$.

(iii) $\Rightarrow$ **(i)** As before, assume that R is not symmetric, i.e. $R(x_0,y_0) \neq R(y_0,x_0)$ for some $x_0, y_0 \in \mathfrak{X}$. Consider the following two cases.

Case 1: Assume $R(x_0,y_0) \leqslant R(y_0,x_0)$. Since $R(x_0,y_0) \neq R(y_0,x_0)$, we have $R(y_0,x_0) \nleqslant R(x_0,y_0)$. Then, by Lemma 2**(ii)** it follows

$$R(y_0,x_0) \to R(x_0,y_0) < 1. \tag{3}$$

For $A = \{y_0\}$, by Lemma 1**(iii)**, $\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes A(z)) = R(x_0,y_0)$, so

$$\begin{aligned}
\underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))(y_0) &= \inf_{x\in\mathfrak{X}}(R(y_0,x) \to \sup_{z\in\mathfrak{X}}(R(x,z) \otimes A(z))) \\
&\leqslant R(y_0,x_0) \to R(x_0,y_0) < 1 = A(y_0)
\end{aligned}$$

by (3). Hence $A \not\subseteq_L \underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))$.

Case 2: Let $R(x_0,y_0) \nleqslant R(y_0,x_0)$. Applying again Lemma 2**(ii)** we get

$$R(x_0,y_0) \to R(y_0,x_0) < 1. \tag{4}$$

For $A = \{x_0\}$ we have $\sup_{z\in\mathfrak{X}}(R(y_0,z) \otimes A(z)) = R(y_0,x_0)$, so

$$\begin{aligned}\underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))(x_0) &= \inf_{y\in\mathfrak{X}}(R(x_0,y) \to \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)))\\ &\leqslant R(x_0,y_0) \to R(y_0,x_0) < 1 = A(x_0)\end{aligned}$$

by (4). Again $A \not\subseteq_L \underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))$.

Again, for properties stated in the above proposition, L need not be commutative.

6.4 Fuzzy Rough Sets in Efas Spaces

In this section we consider L–fuzzy rough sets in approximation spaces with Euclidean fuzzy relations. In these spaces the following properties hold.

Proposition 7. *For any complete, commutative and integral residuated lattice* $(L,\wedge,\vee,\otimes,\to,0,1)$ *and for any L–fuzzy approximation space* Fas$=(\mathfrak{X},R)$ *the following conditions are equivalent:*

(i) Fas $\in$ Efas

(ii) $\overline{\textsc{Fas}}^L(A) \subseteq_L \underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$

(iii) $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L \underline{\textsc{Fas}}_L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$.

Proof.

(i) $\Rightarrow$ **(ii)** Let R be a fuzzy Euclidean relation, i.e. $R(x,y) \otimes R(x,z) \leqslant R(y,z)$ for all $x,y,z \in \mathfrak{X}$. Then for every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$,

$$\begin{aligned}&\underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))(x)\\ &\quad= \inf_{y\in\mathfrak{X}}(R(x,y) \to \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)))\\ &\quad\geqslant \inf_{y\in\mathfrak{X}}(R(x,y) \to \sup_{z\in\mathfrak{X}}((R(x,y) \otimes R(x,z)) \otimes A(z))) && \text{assumption}\\ &\quad= \inf_{y\in\mathfrak{X}}(R(x,y) \to (R(x,y) \otimes \sup_{z\in\mathfrak{X}}(R(x,z) \otimes A(z)))) && \text{Lemma 4}\mathbf{(v)}\\ &\quad\geqslant \inf_{y\in\mathfrak{X}} \sup_{z\in\mathfrak{X}}(R(x,z) \otimes A(z)) && \text{Lemma 1}\mathbf{(v)}\\ &\quad= \overline{\textsc{Fas}}^L(A)(x).\end{aligned}$$

Then $\overline{\textsc{Fas}}^L(A) \subseteq_L \underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A))$.

(ii) $\Rightarrow$ **(i)** Assume that Fas $\notin$ Efas, that is R is not Euclidean. This means that $R(x_0,y_0) \otimes R(x_0,z_0) \nleqslant R(y_0,z_0)$ for some $x_0,y_0,z_0 \in \mathfrak{X}$. By Lemma 2**(ii)** this is equivalent to

$$R(x_0,y_0) \otimes R(x_0,z_0) \to R(y_0,z_0) < 1 \text{ for some } x_0,y_0,z_0 \in \mathfrak{X}. \tag{5}$$

Let us consider $A = \{z_0\}$. By Lemma 1**(iii)**, $\overline{\textsc{Fas}}^L(A)(y) = \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)) = R(y,z_0)$ for any $y \in \mathfrak{X}$. Hence we have:

$$\begin{aligned}
&\overline{\text{FAS}}^{L}(A)(x_0) \to \underline{\text{FAS}}_{L}(\overline{\text{FAS}}^{L}(A))(x_0) \\
&\quad = R(x_0, z_0) \to \inf_{y\in\mathfrak{X}}(R(x_0, y) \to \overline{\text{FAS}}^{L}(A)(y)) \\
&\quad = R(x_0, z_0) \to \inf_{y\in\mathfrak{X}}(R(x_0, y) \to R(y, z_0)) \\
&\quad \leqslant R(x_0, z_0) \to (R(x_0, y_0) \to R(y_0, z_0)) && \text{Lemma 1}\textbf{(ii)} \\
&\quad = R(x_0, z_0) \otimes R(y_0, z_0) \to R(y_0, z_0) && \text{Lemma 2}\textbf{(iii)} \\
&\quad < 1 && \text{from (5).}
\end{aligned}$$

Then, again from Lemma 2**(ii)**, $\overline{\text{FAS}}^{L}(A)(x_0) \nleqslant \underline{\text{FAS}}_{L}(\overline{\text{FAS}}^{L}(A))(x_0)$, so we get $\overline{\text{FAS}}^{L}(A) \not\subseteq_L \underline{\text{FAS}}_{L}(\overline{\text{FAS}}^{L}(A))$.

(i) ⇒ **(iii)** For every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x, y, z \in \mathfrak{X}$,

$$R(y, z) \to A(z) \leqslant (R(x, y) \otimes R(x, z)) \to A(z)$$

by Lemma 1**(ii)**. Next, by Lemma 2**(iii)**,

$$(R(x, y) \otimes R(x, z)) \to A(z) = R(x, y) \to (R(x, z) \to A(z)),$$

so we get for every $A \in \mathcal{F}_L(\mathfrak{X})$ and every $x, y \in \mathfrak{X}$,

$$\begin{aligned}
&\underline{\text{FAS}}_{L}(A)(y) \\
&\quad = \inf_{z\in\mathfrak{X}} \big(R(y, z) \to A(z)\big) \\
&\quad \leqslant \inf_{z\in\mathfrak{X}} \Big(R(x, y) \to \big(R(x, z) \to A(z)\big)\Big) \\
&\quad = R(x, y) \to \inf_{z\in\mathfrak{X}} \big(R(x, z) \to A(z)\big) \\
&\quad = R(x, y) \to \underline{\text{FAS}}_{L}(A)(x)
\end{aligned}$$

by Lemma 4**(ii)**. Therefore, for every $A \in \mathcal{F}_L(\mathfrak{X})$ and every $x \in \mathfrak{X}$,

$$\begin{aligned}
&\overline{\text{FAS}}^{L}(\underline{\text{FAS}}_{L}(A))(x) \\
&\quad = \sup_{y\in\mathfrak{X}}(R(x, y) \otimes \underline{\text{FAS}}_{L}(A)(y)) \\
&\quad \leqslant \sup_{y\in\mathfrak{X}}[R(x, y) \otimes (R(x, y) \to \underline{\text{FAS}}_{L}(A)(x))] \\
&\quad \leqslant \underline{\text{FAS}}_{L}(A)(x)
\end{aligned}$$

by Lemma 1**(vi)**. This implies $\overline{\text{FAS}}^{L}\big(\underline{\text{FAS}}_{L}(A)\big) \subseteq_L \underline{\text{FAS}}_{L}(A)$.

(iii) ⇒ **(i)** Assume that R is not Euclidean, i.e. for some $x_0, y_0, z_0 \in \mathfrak{X}$,

$$R(x_0, y_0) \otimes R(x_0, z_0) \nleqslant R(y_0, z_0).$$

Then, by Lemma 2**(ii)**

$$\big(R(x_0, y_0) \otimes R(x_0, z_0)\big) \to R(y_0, z_0) < 1 \quad \text{for some } x_0, y_0, z_0 \in \mathfrak{X}. \tag{6}$$

Consider $A = y_0R$. Then, by Lemma 4**(i)** we have

$$\begin{aligned}
&\overline{\textsc{Fas}}^{L}(\underline{\textsc{Fas}}_{L}(A))(x_0) \to \underline{\textsc{Fas}}_{L}(A)(x_0)\\
&\quad = [\sup_{y\in\mathfrak{X}}[R(x_0,y)\otimes \inf_{z\in\mathfrak{X}}(R(y,z)\to R(y_0,z))]]\\
&\qquad\qquad \to [\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z))]\\
&\quad = \inf_{y\in\mathfrak{X}}[(R(x_0,y)\otimes \inf_{z\in\mathfrak{X}}(R(y,z)\to R(y_0,z)))\\
&\qquad\qquad \to (\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z)))]\\
&\quad \leqslant (R(x_0,y_0)\otimes \inf_{z\in\mathfrak{X}}(R(y_0,z)\to R(y_0,z)))\\
&\qquad\qquad \to (\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z)))\\
&\quad = (R(x_0,y_0)\otimes 1)\to (\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z)))\\
&\quad = R(x_0,y_0)\to (\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z)))
\end{aligned}$$

by Lemma 2**(ii)** and assumption that L is integral. Next, by Lemma 1**(ii)**,

$$\begin{aligned}
&R(x_0,y_0)\to [\inf_{z\in\mathfrak{X}}(R(x_0,z)\to R(y_0,z))]\\
&\quad \leqslant R(x_0,y_0)\to (R(x_0,z_0)\to R(y_0,z_0)),
\end{aligned}$$

so we have:

$$\begin{aligned}
&\overline{\textsc{Fas}}^{L}(\underline{\textsc{Fas}}_{L}(A))(x_0) \to \underline{\textsc{Fas}}_{L}(A)(x_0)\\
&\quad \leqslant R(x_0,y_0)\to (R(x_0,z_0)\to R(y_0,z_0))\\
&\quad = (R(x_0,y_0)\otimes R(x_0,z_0))\to R(y_0,z_0)\\
&\quad < 1
\end{aligned}$$

by (6) and Lemma 1**(vi)**. Therefore, again by applying Lemma 2**(ii)**, we get $\overline{\textsc{Fas}}^{L}(\underline{\textsc{Fas}}_{L}(A))(x_0) \not\leqslant \underline{\textsc{Fas}}_{L}(A)(x_0)$ so $\overline{\textsc{Fas}}^{L}(\underline{\textsc{Fas}}_{L}(A)) \not\subseteq_L \underline{\textsc{Fas}}_{L}(A)$. ■

6.5 Fuzzy Rough Sets in Tfas Spaces

Proposition 8. *For any complete, commutative and integral residuated lattice* $(L,\wedge,\vee,\otimes,\to,0,1)$ *and for any L–fuzzy approximation space* Fas$=(\mathfrak{X},R)$ *the following conditions are equivalent:*

(i) Fas $\in$ Tfas

(ii) $\underline{\textsc{Fas}}_{L}(A) \subseteq_L \underline{\textsc{Fas}}_{L}(\underline{\textsc{Fas}}_{L}(A))$ *for every* $A\in\mathcal{F}_L(\mathfrak{X})$

(iii) $\overline{\textsc{Fas}}^{L}(\overline{\textsc{Fas}}^{L}(A)) \subseteq_L \overline{\textsc{Fas}}^{L}(A)$ *for every* $A\in\mathcal{F}_L(\mathfrak{X})$.

Proof.

(i)$\Rightarrow$(ii) For every $A\in\mathcal{F}_L(\mathfrak{X})$ and for every $x\in\mathfrak{X}$,

$$\begin{aligned}
&\underline{\textsc{Fas}}_{L}(\underline{\textsc{Fas}}_{L}(A))(x)\\
&= \inf_{y\in\mathfrak{X}}(R(x,y)\to \inf_{z\in\mathfrak{X}}(R(y,z)\to A(z)))\\
&= \inf_{y\in\mathfrak{X}}\inf_{z\in\mathfrak{X}}(R(x,y)\to (R(y,z)\to A(z))) && \text{Lemma 4}\textbf{(ii)}\\
&= \inf_{z\in\mathfrak{X}}\inf_{y\in\mathfrak{X}}(R(x,y)\otimes R(y,z)\to A(z)) && \text{Lemma 2}\textbf{(iii)}\\
&\geqslant \inf_{y\in\mathfrak{X}}(R(x,z)\to A(z)) && \text{assumption and Lemma 1}\textbf{(ii)}\\
&= \underline{\textsc{Fas}}_{L}(A)(x).
\end{aligned}$$

This immediately implies $\underline{\mathrm{FAS}}_L(A) \subseteq_L \underline{\mathrm{FAS}}_L\big(\underline{\mathrm{FAS}}_L(A)\big)$.

(ii) ⇒ (i) Assume that $\mathrm{FAS} \notin \mathrm{TFAS}$. It follows than that for some $x_0, y_0 \in \mathfrak{X}$, $\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y_0)) \nleqslant R(x_0,y_0)$, which by Lemma 2**(ii)** gives

$$(\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y_0))) \rightarrow R(x_0,y_0) < 1 \text{ for some } x_0, y_0 \in \mathfrak{X}. \qquad (7)$$

Consider $A = x_0R$. Note that by Lemma 2**(ii)**,

$$\underline{\mathrm{FAS}}_L(A)(x_0) = \inf_{z\in\mathfrak{X}}(R(x_0,z) \rightarrow R(x_0,z)) = 1.$$

Therefore, by Lemma 2**(i)**,

$$\begin{aligned}&\underline{\mathrm{FAS}}_L(A)(x_0) \rightarrow \underline{\mathrm{FAS}}_L(\underline{\mathrm{FAS}}_L(A))(x_0)\\ &\quad = 1 \rightarrow \inf_{z\in\mathfrak{X}}(R(x_0,z) \rightarrow \inf_{y\in\mathfrak{X}}(R(z,y) \rightarrow R(x_0,y)))\\ &\quad = \inf_{z\in\mathfrak{X}}(R(x_0,z) \rightarrow \inf_{y\in\mathfrak{X}}(R(z,y) \rightarrow R(x_0,y))).\end{aligned}$$

Next, by Lemma 4**(ii)**,

$$\begin{aligned}&\inf_{z\in\mathfrak{X}}(R(x_0,z) \rightarrow \inf_{y\in\mathfrak{X}}(R(z,y) \rightarrow R(x_0,y)))\\ &\quad = \inf_{y\in\mathfrak{X}} \inf_{z\in\mathfrak{X}}(R(x_0,z) \rightarrow (R(z,y) \rightarrow R(x_0,y))).\end{aligned}$$

Furthermore, by Lemma 2**(iii)**,

$$R(x_0,z) \rightarrow (R(z,y) \rightarrow R(x_0,y)) = (R(x_0,z) \otimes R(z,y)) \rightarrow R(x_0,y).$$

Therefore we have:

$$\begin{aligned}&\underline{\mathrm{FAS}}_L(A)(x_0) \rightarrow \underline{\mathrm{FAS}}_L(\underline{\mathrm{FAS}}_L(A))(x_0)\\ &\quad = \inf_{y\in\mathfrak{X}} \inf_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y) \rightarrow R(x_0,y))\\ &\quad = \inf_{y\in\mathfrak{X}}(\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y)) \rightarrow R(x_0,y))\end{aligned}$$

by Lemma 4**(i)**. Finally, by (7) we have:

$$\begin{aligned}&\inf_{y\in\mathfrak{X}}(\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y)) \rightarrow R(x_0,y))\\ &\quad \leqslant (\sup_{y\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y_0))) \rightarrow R(x_0,y_0) < 1,\end{aligned}$$

whence $\underline{\mathrm{FAS}}_L(A)(x_0) \rightarrow \underline{\mathrm{FAS}}_L(\underline{\mathrm{FAS}}_L(A))(x_0) < 1$. By Lemma 2**(ii)** this implies $\underline{\mathrm{FAS}}_L(A)(x_0) \nleqslant \underline{\mathrm{FAS}}_L(\underline{\mathrm{FAS}}_L(A))(x_0)$.

Then $\underline{\mathrm{FAS}}_L(A) \nsubseteq_L \underline{\mathrm{FAS}}_L(\underline{\mathrm{FAS}}_L(A))$.

(i) ⇒ (iii) For every $A \in \mathcal{F}_L(\mathfrak{X})$ and for every $x \in \mathfrak{X}$,

$$\begin{aligned}&\overline{\mathrm{FAS}}^L(\overline{\mathrm{FAS}}^L(A))(x)\\ &= \sup_{y\in\mathfrak{X}}(R(x,y) \otimes \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)))\\ &= \sup_{z\in\mathfrak{X}} \sup_{y\in\mathfrak{X}}(R(x,y) \otimes R(y,z) \otimes A(z)) && \text{Lemma 4}\textbf{(v)}\\ &\leqslant \sup_{z\in\mathfrak{X}} \sup_{y\in\mathfrak{X}}(R(x,z) \otimes A(z))) && \text{assumption and Lemma 1}\textbf{(i)}\\ &= \overline{\mathrm{FAS}}^L(A)(x).\end{aligned}$$

Then $\overline{\mathrm{FAS}}^L(\overline{\mathrm{FAS}}^L(A)) \subseteq_L \overline{\mathrm{FAS}}^L(A)$.

(iii) $\Rightarrow$ **(i)** Assume that $\textsc{Fas} \notin \textsc{Tfas}$, i.e. R is not transitive. This means that for some $x_0, y_0 \in \mathfrak{X}$, $\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y_0)) \nleq R(x_0,y_0)$. By Lemma 2**(ii)**,

$$\sup_{z\in\mathfrak{X}}(R(x_0,z) \otimes R(z,y_0)) \rightarrow R(x_0,y_0) < 1. \tag{8}$$

For $A = \{z_0\}$ we have $\overline{\textsc{Fas}}^L(A)(y) = \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)) = R(y,z_0)$ for every $y \in \mathfrak{X}$, so

$$\begin{aligned}
&\overline{\textsc{Fas}}^L(\overline{\textsc{Fas}}^L(A))(x_0) \rightarrow \overline{\textsc{Fas}}^L(A)(x_0)\\
&\quad = [\sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes \sup_{z\in\mathfrak{X}}(R(y,z) \otimes A(z)))] \rightarrow R(x_0,z_0)\\
&\quad = [\sup_{z\in\mathfrak{X}} \sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(y,z) \otimes A(z))] \rightarrow R(x_0,z_0)\\
&\quad = [\sup_{z\in\mathfrak{X}}(\sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(y,z)) \otimes A(z))] \rightarrow R(x_0,z_0)\\
&\quad = [\sup_{y\in\mathfrak{X}}(R(x_0,y) \otimes R(y,z_0))] \rightarrow R(x_0,z_0)\\
&\quad < 1
\end{aligned}$$

by (8). Then $\overline{\textsc{Fas}}^L(\overline{\textsc{Fas}}^L(A))(x_0) \nleq \overline{\textsc{Fas}}^L(A)(x_0)$, which immediately implies $\overline{\textsc{Fas}}^L(\overline{\textsc{Fas}}^L(A)) \not\subseteq_L \overline{\textsc{Fas}}^L(A)$. ■

6.6 Fuzzy Rough Sets in Specific Fuzzy Approximation Spaces

We conclude the chapter by giving properties of fuzzy rough sets in some more specific classes of fuzzy approximation spaces.

Corollary 2. *For any complete, commutative and integral residuated lattice* $(L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *and for any L–fuzzy approximation space* $\textsc{Fas}=(\mathfrak{X}, R)$ *the following conditions hold:*

(i) *R is reflexive and transitive iff*
- $\overline{\textsc{Fas}}^L(\overline{\textsc{Fas}}^L(A)) = \overline{\textsc{Fas}}^L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$
- $\underline{\textsc{Fas}}_L(\underline{\textsc{Fas}}_L(A)) = \underline{\textsc{Fas}}_L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$

(ii) *if R is an L–equivalence relation then*
- $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) = \underline{\textsc{Fas}}_L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$
- $\underline{\textsc{Fas}}_L(\overline{\textsc{Fas}}^L(A)) = \overline{\textsc{Fas}}^L(A)$ *for every* $A \in \mathcal{F}_L(\mathfrak{X})$.

Proof. **(i)** Results from Propositions 5 and 8.

(ii) By transitivity of R we get from Proposition 8

$$\underline{\textsc{Fas}}_L(A) \subseteq_L \underline{\textsc{Fas}}_L(\underline{\textsc{Fas}}_L(A)) \tag{9}$$

Next, reflexivity of R implies

$$\underline{\textsc{Fas}}_L(A) \subseteq_L \overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \quad \text{for every } A \in \mathcal{F}_L(\mathfrak{X}). \tag{10}$$

Furthermore, from (9) and Proposition 2**(ii)**,

$$\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L \overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(\underline{\textsc{Fas}}_L(A))). \tag{11}$$

Finally, symmetry of R implies by Proposition 6,

$$\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(\underline{\textsc{Fas}}_L(A))) \subseteq_L \underline{\textsc{Fas}}_L(A). \tag{12}$$

(11)–(12) imply $\overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A)) \subseteq_L \underline{\textsc{Fas}}_L(A)$ which by (10) gives for every $A \in \mathcal{F}_L(\mathfrak{X})$, $\underline{\textsc{Fas}}_L(A) = \overline{\textsc{Fas}}^L(\underline{\textsc{Fas}}_L(A))$.

The second equality can be proved in the analogous way. ■

7 Conclusions

In this paper, we have generalized the notion of fuzzy rough sets taking an arbitrary residuated lattice as a basic algebraic structure. Several classes of these L–fuzzy rough sets have been considered and basic properties of these classes have been investigated.

This paper gives a basis for further investigations. The theory of rough sets gave rise to a wide methodology for analyzing incompleteness of information in information systems. It is therefore natural to consider hybrid fuzzy–rough systems. In particular, an interesting problem concerns fuzzy decision rules: how to combine methods commonly applied in fuzzy logics with reasoning methods traditionally developed within the framework of rough set theory.

Moreover, as we have pointed out, fuzzy rough approximators correspond to the basic fuzzy modal operators Having established main properties of particular classes of fuzzy rough sets, we can formulate characteristic axioms for the corresponding systems of fuzzy modal logics. The problem of axiomatization of these logics is, in general, an open problem.

These problems are investigated in our ongoing papers.

Acknowledgements

The work was carried out in the framework of COST Action 274 on *Theory and Applications of Relational Structures as Knowledge Instruments.*

References

1. R. P. Dilworth, N. Ward. Residuated lattices. *Trans. Amer. Math. Soc.* **45**, 1939, pp. 335–354.
2. D. Dubois, H. Prade. Rough fuzzy sets and fuzzy rough sets. *Int. J. of General Systems* **17: 2-3** , 1990, pp. 191–209.
3. D. Dubois, H. Prade. Putting fuzzy sets and rough sets together, in: *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory* (R. Słowiński (ed.)). Kluwer, 1992, pp. 203–232.
4. F. Esteva, L. Godo. . Monoidal t–norm based logic: towards a logic for left–continuous t–norms. *Fuzzy Sets and Systems* **124**, 2001, pp. 271–288.
5. J. A. Goguen. (1967). L–fuzzy sets. *Journal of Mathematical Analysis and Applications* **18**, pp. 145–174.

6. P. Hajek . *Metamathematics of Fuzzy Logics*, Kluwer, 1998.
7. A. Nakamura. Fuzzy rough sets. *Notes on Multiple–Valued Logic in Japan* **9: 8**, 1988, pp. 1–8.
8. S. Nanda, S. Majumdar Fuzzy rough sets. *Fuzzy Sets and Systems* **45**, 1992, pp. 157–160.
9. E. Orłowska (ed.) . *Incomplete Information: Rough Set Analysis.* Springer, 1998.
10. S. K. Pal, A. Skowron. *Rough Fuzzy Hybridization: A New Trend in Decision Making.* Springer Singapore, 1999.
11. Z. Pawlak. Rough sets. *Int. Journal of Computer and Information Science* **11: 5**, 1982, pp. 341–356.
12. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer, 1991.
13. L. Polkowski, A. Skowron. *Rough Sets and Knowledge Discovery* vols. 1, 2. Physica, 1998.
14. A. M. Radzikowska, E. E. Kerre. Fuzzy rough sets revisited, in: *Proceedings EUFIT'99*, Aachen, 1999, (published on CD).
15. A. M. Radzikowska, E. E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems* **126**, 2001, pp. 137–155.
16. A. M. Radzikowska, E. E. Kerre. A general calculus of fuzzy rough sets. Submitted.
17. H. Rasiowa, R. Sikorski. *The Mathematics of Metamathematics.* PWN – Polish Scientific Publishers, 1970.
18. R. Słowiński (ed.). *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory.* Kluwer, 1992.
19. H. Thiele. On the definition of modal operators in fuzzy logic, in: *Proceedings ISMVL-93*, Sacramento, California, 1993, pp. 62–67.
20. H. Thiele. *Fuzzy Rough Sets versus Rough Fuzzy Sets – An Interpretation and a Comparative Study using Concepts of Modal Logics.* Technical Report No CI-30/98, University of Dortmund, 1998. Also in *Proceedings EUFIT'97*, Aachen, 1997, pp. 159–167.
21. H. Thiele. Generalizing the explicit concept of rough set on the basis of modal logic. To appear in: *Computational Intelligence in Theory and Practice.* Advances in Soft Computing, Physica.
22. E. Turunen. *Mathematics Behind Fuzzy Logics.* Physica, 1999.
23. L. A. Zadeh. Fuzzy sets. *Information and Control* **8**, 1965, pp. 338–358.
24. W. P. Ziarko (ed.). *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer London, 1994.

Semantics of Fuzzy Sets in Rough Set Theory

Yiyu Yao

Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
yyao@cs.uregina.ca
www.cs.uregina.ca/~yyao

Abstract. The objective of this paper is to provide a semantic framework for fuzzy sets in the theory of rough sets. Rough membership functions are viewed as a special type of fuzzy membership functions interpretable using conditional probabilities. The relationships between fuzzy membership functions and rough membership functions, between core and support of fuzzy set theory and lower and upper approximation of rough set theory, are investigated. It is demonstrated that both theories share the same qualitative properties. Interpretations of fuzzy sets in rough set theory lead to constraints on membership values. Two types of constraints on membership values are studies, namely, constraints on membership values of related elements and constraints on membership values of related sets. The classical rough set model and generalized rough set models are discussed.

1 Introduction

The theory of rough sets deals with the approximation of sets under indiscernibility [7, 8]. The theory of fuzzy sets handles vague concepts by allowing partial memberships [23]. Both theories are motivated by the practical needs to manage and process uncertainty inherent in real world problem solving. They are different and complementary generalizations of classical set theory [2, 10, 14].

Semantic interpretation of a fuzzy membership grade is one of the fundamental issues of fuzzy set theory [4]. Dubois and Prade examined three main semantics of membership functions, namely, similarity, preference and uncertainty [3, 4]. Each of them underlies a particular class of applications. The co-existence of multiple interpretations shows the richness of the concept of fuzzy sets.

The similarity based semantics relies on relationships between objects of a universe in terms of a similarity, a dissimilarity, or a distance function. The membership of an object is the degree of proximity of the object to the prototype elements of a fuzzy set [4]. The similarity based semantics imposes certain qualitative and quantitative constraints, reflecting the similarity of objects. The more similar an object is to the prototype elements of a fuzzy set, the greater is its membership grade, and vice versa. The membership grades can be computed

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 297–318, 2004.

by means of an increasing function of the similarity between an object and the prototype elements of a fuzzy set [3, 4].

The theory of rough sets is based on a well studied and practical notion called equivalence relations. Equivalence relations are defined and explained in terms of an information table, such as a database. By focusing on one particular type of uncertainty, namely, the discernibility or indistinguishability of elements, rough set theory has a systematic treatment of its semantic interpretations. The formulation and interpretation of rough sets as defined by rough membership functions, set-theoretic operators, as well as rough set approximation operators are inseparable parts of the theory.

Equivalence relations describe a specific type of similarity between objects. Other types of binary relations are also used in generalizations of rough sets [20]. A question arises naturally is whether we can adopt the methodologies used in the study of rough sets to investigate fuzzy sets. We attempt to provide a positive answer to the question in this paper. Similarity based semantics of fuzzy sets is discussed in terms of rough sets. While many investigations of fuzzy sets focus on quantitative study of the theory, we focus on the qualitative aspects of the theory. Rough membership functions are viewed as a special type of fuzzy membership functions interpretable using conditional probabilities. Rough set approximations are related to the core and support of a fuzzy set. Based on these connections, we demonstrate that both theories share the same qualitative features. We also study qualitative constraints on membership values and approximation operators based on discernibility and similarity of elements. The constraints and approximation operators are studied in detail with respect to the classical rough set model based on equivalence relations or partitions, generalized rough set models based on arbitrary binary relations, and generalized rough set models based on coverings [15].

2 A Critical Review of Fuzzy Sets

The notion of fuzzy sets provides a convenient tool for representing vague concepts by allowing partial memberships [23]. Depending on the characteristics of membership values, two types of fuzzy sets have been studied [6]. If fuzzy membership functions take a single value from the unit interval [0,1] or a lattice, the corresponding fuzzy sets are known as type 1 fuzzy sets. If fuzzy membership functions take type 1 fuzzy sets as their values, the corresponding fuzzy sets are known as type 2 fuzzy sets.

In this paper, we restrict our discussion to the classical fuzzy sets, i.e., the type 1 fuzzy sets. Furthermore, we assume that the universe, namely, the referential set on which fuzzy sets are defined, is finite. For simplicity, we use the same symbols for both standard set-theoretic relations and operators and fuzzy set-theoretic relations and operators.

2.1 Fuzzy Set Systems

Let U be a finite and non-empty set called universe. A fuzzy subset of U is defined by a membership function:

$$\mu_{\mathcal{A}} : U \longrightarrow [0,1]. \tag{1}$$

Fuzzy set inclusion and equality can be defined component-wise. A fuzzy set $\mu_{\mathcal{A}}$ is a subset of another fuzzy set $\mu_{\mathcal{B}}$, written $\mu_{\mathcal{A}} \subseteq \mu_{\mathcal{B}}$ if and only if $\mu_{\mathcal{A}}(x) \leq \mu_{\mathcal{B}}(x)$ for all $x \in U$. Fuzzy set $\mu_{\mathcal{A}}$ is equal to fuzzy set $\mu_{\mathcal{B}}$, written $\mu_{\mathcal{A}} = \mu_{\mathcal{B}}$ if and only if $\mu_{\mathcal{A}}(x) = \mu_{\mathcal{B}}(x)$ for all $x \in U$. Obviously, $\mu_{\mathcal{A}} = \mu_{\mathcal{B}}$ if and only if $\mu_{\mathcal{A}} \subseteq \mu_{\mathcal{B}}$ and $\mu_{\mathcal{B}} \subseteq \mu_{\mathcal{A}}$.

There are many definitions for fuzzy set complement, intersection, and union. The standard min-max system proposed by Zadeh is given component-wise by [23]:

$$\begin{aligned} \mu_{\neg\mathcal{A}}(x) &= 1 - \mu_{\mathcal{A}}(x), \\ \mu_{\mathcal{A}\cap\mathcal{B}}(x) &= \min(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)), \\ \mu_{\mathcal{A}\cup\mathcal{B}}(x) &= \max(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)). \end{aligned} \tag{2}$$

A crisp subset of U may be viewed as a degenerated fuzzy set. In this case, the membership function is the characteristic function taking only two values 0 and 1. The min-max fuzzy set-theoretic operators reduce to the classical set-theoretic operators when characteristic functions are used.

In general, one may define fuzzy set operators using triangular norms (t-norms) and conorms (t-conorms) [1, 5, 6]. A t-norm is a function from $[0,1]\times[0,1]$ to $[0,1]$ and satisfies the following conditions: for $a, b, c \in [0,1]$,

(i). Boundary conditions
$t(0,0) = 0,$
$t(1,a) = t(a,1) = a;$

(ii). Monotonicity
$(a \leq c, b \leq d) \Longrightarrow t(a,b) \leq t(c,d);$

(iii). Symmetry
$t(a,b) = t(b,a);$

(iv). Associativity
$t(a, t(b,c)) = t(t(a,b), c)).$

For clarity, some of the conditions are explicitly listed, although they can be obtained from other conditions. For example, the symmetry implies $t(1,a) = t(a,1)$. We only need either $t(a,1) = a$ or $t(1,a) = a$. The condition $t(0,0) = 0$ follows from $t(1,a) = a$, the symmetry and the monotonicity. The boundary conditions ensure that a t-norm reduces to standard set intersection when only crisp sets are involved. The monotonicity suggests that fuzzy set intersection keep the monotonicity of set intersection with respect to set inclusion. Another version of monotonicity is given by $b \leq c \Longrightarrow t(a,b) \leq t(a,c)$.

Some commonly used t-norms are $t_{\mathrm{b}}(a,b) = \max(0, a+b-1)$, $t_{\min}(a,b) = \min(a,b)$, the product operator $t_{\mathrm{p}}(a,b) = ab$, and t_{w} defined by boundary conditions and $t_{\mathrm{w}}(a,b) = 0, \forall (a,b) \in [0,1) \times [0,1)$. These t-norms are related by inequality [1, 6]:

$$t_{\mathrm{w}}(a,b) \leq t_{\mathrm{b}}(a,b) \leq t_{\mathrm{p}}(a,b) \leq t_{\min}(a,b). \tag{3}$$

Moreover, any t-norm is bounded by t_{w} and $t_{\min}$, i.e.,

$$t_{\mathrm{w}}(a,b) \leq t(a,b) \leq t_{\min}(a,b). \tag{4}$$

It provides the bounds of the membership values of the intersection of two fuzzy sets.

Suppose $n : [0,1] \longrightarrow [0,1]$ is an operator called negation. With respect to a negation operator, the dual of a t-norm is given by $n(t(n(a), n(b)))$ and called a t-conorm, which is a function s mapping $[0,1] \times [0,1]$ to $[0,1]$ and satisfying the boundary conditions:

$$\begin{aligned} \text{(i}'\text{).}\quad & \text{Boundary conditions} \\ & s(1,1) = 1, \\ & s(a,0) = s(0,a) = a, \end{aligned}$$

and conditions of monotonicity, symmetry, and associativity. Suppose the negation operator is defined by $n(a) = 1 - a$. The t-conorm s corresponding to a t-norm t is given by:

$$\begin{aligned} s(a,b) &= n(t(n(a), n(b))) \\ &= 1 - t(1-a, 1-b). \end{aligned} \tag{5}$$

The t-conorms of $t_{\min}$, t_{p} and t_{b} are $s_{\max}(a,b) = \max(a,b)$, $s_{\mathrm{p}}(a,b) = a + b - ab$, and $s_{\mathrm{b}}(a,b) = \min(1, a+b)$, respectively. The t-conorm corresponding to t_{w} is given by the boundary conditions and $s_{\mathrm{w}}(a,b) = 1, \forall (a,b) \in (0,1] \times (0,1]$. Similarly, we have:

$$s_{\max}(a,b) \leq s_{\mathrm{p}}(a,b) \leq s_{\mathrm{b}}(a,b) \leq s_{\mathrm{w}}(a,b). \tag{6}$$

Any t-conorm is bounded by $s_{\max}$ and s_{w}:

$$s_{\max}(a,b) \leq s(a,b) \leq s_{\mathrm{w}}(a,b). \tag{7}$$

It provides the bounds of the membership values of the union of two fuzzy sets. Combining with equation (4), we have:

$$t(a,b) \leq \min(a,b) \leq \max(a,b) \leq s(a,b), \tag{8}$$

which expresses the connection between fuzzy set intersection and union.

Let t and s be a pair of t-norm and t-conorm. We define fuzzy set intersection and union component-wise by:

$$\begin{aligned} \mu_{\mathcal{A} \cap \mathcal{B}}(x) &= t(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)), \\ \mu_{\mathcal{A} \cup \mathcal{B}}(x) &= s(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)). \end{aligned} \tag{9}$$

An important feature of fuzzy set operators as defined by t-norms and t-conorms is that they are truth-functional operators. In other words, membership functions of complement, intersection, and union of fuzzy sets are defined based solely on the membership functions of the fuzzy sets involved [14].

2.2 Qualitative Characterization of Fuzzy Sets

Although the conditions on t-norms and t-conorms are expressed in quantitative forms, they are of qualitative nature that characterize a large class of functions. From t-norms, one can easily obtain the following qualitative conditions:

$$
\begin{array}{rl}
\text{(I).} & \text{Boundary conditions} \\
& \mu_{\emptyset \cap \emptyset} = \emptyset, \\
& \mu_{U \cap \mathcal{A}} = \mu_{\mathcal{A} \cap U} = \mu_{\mathcal{A}}, \\
\text{(II).} & \text{Monotonicity} \\
& [\mu_{\mathcal{A}} \subseteq \mu_{\mathcal{C}}, \mu_{\mathcal{B}} \subseteq \mu_{\mathcal{D}}] \Longrightarrow \mu_{\mathcal{A} \cap \mathcal{B}} \subseteq \mu_{\mathcal{C} \cap \mathcal{D}}, \\
\text{(III).} & \text{Commutativity} \\
& \mu_{\mathcal{A} \cap \mathcal{B}} = \mu_{\mathcal{B} \cap \mathcal{A}}, \\
\text{(IV).} & \text{Associativity} \\
& \mu_{(\mathcal{A} \cap \mathcal{B}) \cap \mathcal{C}} = \mu_{\mathcal{A} \cap (\mathcal{B} \cap \mathcal{C})}.
\end{array}
$$

For the t-conorms, we have:

$$
\begin{array}{rl}
\text{(I}'\text{).} & \text{Boundary conditions} \\
& \mu_{U \cup U} = U, \\
& \mu_{\emptyset \cup \mathcal{A}} = \mu_{\mathcal{A} \cup \emptyset} = \mu_{\mathcal{A}}, \\
\text{(II}'\text{).} & \text{Monotonicity} \\
& [\mu_{\mathcal{A}} \subseteq \mu_{\mathcal{C}}, \mu_{\mathcal{B}} \subseteq \mu_{\mathcal{D}}] \Longrightarrow \mu_{\mathcal{A} \cup \mathcal{B}} \subseteq \mu_{\mathcal{C} \cup \mathcal{D}}, \\
\text{(III}'\text{).} & \text{Commutativity} \\
& \mu_{\mathcal{A} \cup \mathcal{B}} = \mu_{\mathcal{B} \cup \mathcal{A}}, \\
\text{(IV}'\text{).} & \text{Associativity} \\
& \mu_{(\mathcal{A} \cup \mathcal{B}) \cup \mathcal{C}} = \mu_{\mathcal{A} \cup (\mathcal{B} \cup \mathcal{C})}.
\end{array}
$$

Compared with the quantitative conditions on t-norms and t-conorms, the qualitative properties are easier to interpret. By the boundary conditions and monotonicity, we have:

$$\mu_{\mathcal{A} \cap \mathcal{B}} \subseteq \mu_{\mathcal{A}} \subseteq \mu_{\mathcal{A} \cup \mathcal{B}}, \tag{10}$$

which corresponds to the quantitative condition in equation (8).

The qualitative conditions are much weaker than the quantitative conditions. Under the qualitative conditions, fuzzy set intersection does not have to be truth-functional. If we assume the truth-functionality of fuzzy set operators, we can obtain the quantitative properties from the qualitative ones.

The concepts of core and support have been introduced and used as approximations of a fuzzy set [6]. The core of a fuzzy set $\mathcal{A}$ is a crisp subset of U consisting of elements with full membership:

$$\text{core}(\mu_{\mathcal{A}}) = \{x \in U \mid \mu_{\mathcal{A}}(x) = 1\}. \tag{11}$$

The support is a crisp subset of U consisting of elements with non-zero membership:

$$\text{support}(\mu_{\mathcal{A}}) = \{x \in U \mid \mu_{\mathcal{A}}(x) > 0\}. \tag{12}$$

With $1 - (\cdot)$ as fuzzy set complement, and a pair of dual t-norm and t-conorm as fuzzy set intersection and union, the following properties hold:

$$\begin{array}{ll}
\text{(F1)} & \text{core}(\mu_{\mathcal{A}}) = \neg(\text{support}(\mu_{\neg\mathcal{A}})), \\
 & \text{support}(\mu_{\mathcal{A}}) = \neg(\text{core}(\mu_{\neg\mathcal{A}})), \\
\text{(F2)} & \text{core}(\mu_{\mathcal{A}\cap\mathcal{B}}) = \text{core}(\mu_{\mathcal{A}}) \cap \text{core}(\mu_{\mathcal{B}}), \\
 & \text{support}(\mu_{\mathcal{A}\cap\mathcal{B}}) \subseteq \text{support}(\mu_{\mathcal{A}}) \cap \text{support}(\mu_{\mathcal{B}}), \\
\text{(F3)} & \text{core}(\mu_{\mathcal{A}\cup\mathcal{B}}) \supseteq \text{core}(\mu_{\mathcal{A}}) \cup \text{core}(\mu_{\mathcal{B}}), \\
 & \text{support}(\mu_{\mathcal{A}\cup\mathcal{B}}) = \text{support}(\mu_{\mathcal{A}}) \cup \text{support}(\mu_{\mathcal{B}}), \\
\text{(F4)} & \text{core}(\mu_{\mathcal{A}}) \subseteq \mu_{\mathcal{A}} \subseteq \text{support}(\mu_{\mathcal{A}}).
\end{array}$$

Property (F1) easily follows from the definition of negation operator $n(a) = 1-a$. The core of a fuzzy set is the complement of the support of the complement of the fuzzy set, and vice versa. Properties (F2) and (F3) follow from the properties of t-norm and t-conorm. The core of fuzzy set intersection can be obtained from the intersection of the cores of two fuzzy sets, and the support of fuzzy set union can be obtained from the union of the supports of two fuzzy sets. However, the support of fuzzy set intersection and the core of fuzzy set union cannot be thus obtained. Property (F4) suggests that a fuzzy set lies within its core and support.

By the boundary conditions of t-norms and t-conorms, we have:

$$\begin{array}{l}
(\text{support}(\mu_{\mathcal{A}}) \cap \text{core}(\mu_{\mathcal{B}})) \cup (\text{core}(\mu_{\mathcal{A}}) \cap \text{support}(\mu_{\mathcal{B}})) \subseteq \text{support}(\mu_{\mathcal{A}\cap\mathcal{B}}), \\
\text{core}(\mu_{\mathcal{A}\cup\mathcal{B}}) \subseteq (\text{support}(\mu_{\mathcal{A}}) \cup \text{core}(\mu_{\mathcal{B}})) \cap (\text{core}(\mu_{\mathcal{A}}) \cup \text{support}(\mu_{\mathcal{B}})).
\end{array}$$

When the min-max system, or the (t_{p}, s_{p}) system, is used, the subset relationships become equality in (F2) and (F3), namely:

$$\begin{array}{ll}
\text{(F2}'\text{)} & \text{support}(\mu_{\mathcal{A}\cap\mathcal{B}}) = \text{support}(\mu_{\mathcal{A}}) \cap \text{support}(\mu_{\mathcal{B}}), \\
\text{(F3}'\text{)} & \text{core}(\mu_{\mathcal{A}\cup\mathcal{B}}) = \text{core}(\mu_{\mathcal{A}}) \cup \text{core}(\mu_{\mathcal{B}}).
\end{array}$$

In order to have properties (F2′) and (F3′), we need the following additional properties on t-norms and t-conorms:

$$\begin{array}{ll}
\text{(v).} & [a \neq 0, b \neq 0] \Longrightarrow t(a,b) \neq 0; \\
\text{(v}'\text{).} & [a \neq 1, b \neq 1] \Longrightarrow s(a,b) \neq 1.
\end{array}$$

Unlike the standard properties of t-norms and t-conorms, they cannot be easily expressed in qualitative terms.

3 A Semantic Framework for Fuzzy Sets

In the previous discussion, we focused on the qualitative properties of fuzzy set theory. Successful applications of fuzzy sets depend on, to a large extent, the interpretations of various notions of the theory. In this section, we present a semantic framework for the interpretation of fuzzy membership values, which is closely related to rough set theory [18].

When applying the theory of fuzzy sets, we do not deal with an abstract notion of sets. More often than not, the referential set U contains objects relevant to a particular problem. Furthermore, we must have some information or knowledge about the objects under consideration. Such knowledge enables us to derive a more concrete and operational theory of fuzzy sets. Within the restricted theory, it is possible to discuss semantic interpretations of membership values.

In order to build a semantic framework for fuzzy sets, we consider the following two levels of knowledge about the universe:

Representations of objects: It is assumed that elements of the universe U are not abstract entities, but physical objects described in terms of a set of attributes, properties or features. We perceive these objects through their features.

Relationships between objects: It is assumed that the available knowledge and information is sufficient for us to construct a certain structure on the universe U. That is, U contains related and interconnected, rather than isolated, elements.

The two assumptions are reasonable. For solving any real world problem, we must have at least some information about the objects under consideration. We should be able to observe, measure, or describe the objects based on their properties. The fact that we consider a set of objects U together implies that they must be somehow related.

At a more concrete level, we assume that each attribute has a set of values. One can observe certain relationships between attribute values. For example, one can use the trivial equality relation. From the first level knowledge, we can build the second level knowledge. The theory of rough set is developed based on the two levels of knowledge [8]. The first level of knowledge is represented by an information table. The second level of knowledge is represented by an equivalence relation on the universe. The equivalence relation on U can be defined with respect to an information table.

Let K represent the available information and knowledge about the universe. In contrast to the abstract fuzzy set theory, we consider the pair (U, K) in an operational fuzzy set theory. The knowledge K provides a basis for the definition of fuzzy sets and fuzzy set-theoretic operators. More specifically, we are interested in a constructive theory of fuzzy sets, in which fuzzy sets are explicitly constructed based on the knowledge K. A similar structure is used by Zhang and Zhang for the study of a granular computing approach for problem solving [25].

A constructive fuzzy set theory uses a triplet (U, K, M), where M refers to a method by which fuzzy sets are constructed. Since the construction of fuzzy sets is an inseparatable part of the theory, one can easily provide a semantic interpretation of fuzzy sets.

In the rest of this paper, the proposed semantic framework is used to interpret fuzzy sets in rough set theory. The knowledge K is given in terms of equivalence relations (partitions), non-equivalence relations, and coverings of U. The method M is discussed according to different types of knowledge. This may provide a possible solution to a fundamental difficulty with fuzzy set theory regarding semantic interpretations of fuzzy membership values.

4 Classical Rough Membership Functions and Rough Set Approximations

The notion of indiscernibility provides a formal way to describe the relationships between elements of a universe. Two elements are said to be indiscernible, indistinguishable, or equivalent if one cannot differentiate them based on the available information or knowledge. The theory of rough sets is a model of uncertainty developed based on the notion of indiscernibility [7, 14, 16].

4.1 Rough Membership Functions

Let $E \subseteq U \times U$ be an equivalence relation on a finite and non-empty universe U. That is, E is reflexive, symmetric, and transitive. The relation E induces a partition U/E of the universe U, i.e., a family of disjoint subsets of the universe known as equivalence classes. The equivalence relation represents the available knowledge about the universe. In the presence of indiscernibility, we are forced to consider an equivalence class as a whole instead of individuals. It is therefore expected that any model of uncertainty should take into consideration the equivalence or indiscernibility of elements.

The pair, $apr = (U, E)$, is called an approximation space. An element $x \in U$ belongs to one and only one equivalence class. Let

$$[x]_E = \{y \mid xEy\}, \tag{13}$$

denote the equivalence class containing x. For a subset $A \subseteq U$, we define a rough membership function [9, 11, 21]:

$$\mu_A(x) = \frac{|[x]_E \cap A|}{|[x]_E|}, \tag{14}$$

where $|\cdot|$ denotes the cardinality of a set. The rough membership value $\mu_A(x)$ may be interpreted as the conditional probability that an arbitrary element belongs to A given that the element belongs to $[x]_E$. The set A is called a generating set of the rough membership μ_A.

Rough membership functions may be interpreted as fuzzy membership functions in terms of the probabilities defined simply by the cardinalities of sets.

With this interpretation, one can define at most $2^{|U|}$ fuzzy sets. Two distinct subsets of U may derive the same rough membership function. By definition, the membership values are all rational numbers.

The theory of fuzzy sets is typically developed as an uninterpreted mathematical theory of abstract membership functions without the above limitations [6]. In contrast, the theory of rough set provides a more specific and more concrete interpretation of fuzzy membership functions. The source of the fuzziness in describing a concept is the indiscernibility of elements. The limitations and constraints of such an interpreted sub-theory should not be viewed as the disadvantages of the theory. In fact, such constraints suggest conditions that can be verified when applying the theory to real world problems. It might be more instructive and informative if one knows that a certain theory cannot be applied. Explicit statements of conditions under which a particular model is applicable may prevent misuses of the theory.

When interpreting fuzzy membership functions in the theory of rough sets, we have the constraints:

$$\begin{array}{ll}
\text{(rm1)} & \mu_U(x) = 1, \\
\text{(rm2)} & \mu_\emptyset(x) = 0, \\
\text{(rm3)} & xEy \Longrightarrow \mu_A(x) = \mu_A(y), \\
\text{(rm4)} & x \in A \Longrightarrow \mu_A(x) \neq 0 \\
 & \mu_A(x) = 0 \Longrightarrow x \notin A, \\
\text{(rm5)} & \mu_A(x) = 1 \Longrightarrow x \in A, \\
 & x \notin A \Longrightarrow \mu_A(x) \neq 1, \\
\text{(rm6)} & A \subseteq B \Longrightarrow \mu_A(x) \leq \mu_B(x).
\end{array}$$

Property (rm3) is particularly important. It shows one type of constraints on membership values. Elements in the same equivalence class must have the same degree of membership. That is, indiscernible elements should have the same membership value. Such a constraint, which ties the membership values of individual elements according to their connections, is intuitively appealing. The two properties in (rm4) are equivalent. An element in A must have a non-zero membership, while an element with zero membership cannot be in A. Similarly, properties in (rm5) state that an element with full membership must be in A, and an element not in A cannot have a full membership. Property (rm6) shows another type of constraints on membership values. A subset of a set generates a fuzzy subset of the corresponding fuzzy set generated by that set.

The constraints on rough membership functions have significant implications on rough set-theoretic operators. Rough membership functions corresponding to $\neg A$, $A \cap B$, and $A \cup B$ must be defined using set-theoretic operators and equation (14). By laws of probability, we have:

$$\begin{array}{ll}
\text{(o1)} & \mu_{\neg A}(x) = 1 - \mu_A(x), \\
\text{(o2)} & \mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_{A\cap B}(x), \\
 & A \cap B = \emptyset \Longrightarrow \mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x),
\end{array}$$

$$\begin{array}{ll}\text{(o3)} & \max(0, \mu_A(x) + \mu_B(x) - 1) \leq \mu_{A\cap B}(x) \leq \min(\mu_A(x), \mu_B(x)),\\ \text{(o4)} & \max(\mu_A(x), \mu_B(x)) \leq \mu_{A\cup B}(x) \leq \min(1, \mu_A(x) + \mu_B(x)).\end{array}$$

Unlike the commonly used fuzzy set operators as typically defined by t-norms and t-conorms, the new intersection and union operators are non-truth-functional. That is, it is impossible to obtain rough membership functions of $A \cap B$ and $A \cup B$ based solely on the rough membership functions of A and B. One must also consider their overlap and their relationships to the equivalence class $[x]_E$.

By property (o2), we have the following formula to calculate the membership function of a set:

$$\mu_A(x) = \sum_{y\in A} \mu_{\{y\}}(x). \tag{15}$$

Thus, rough membership functions corresponding to singleton subsets of U play an important role. They serve as a basis of rough membership functions generated by non-singleton subsets.

4.2 Rough Set Approximations and Approximation Operators

In an approximation space, a subset $A \subseteq U$ is approximated by a pair of sets called the lower and upper approximations [9]:

$$\begin{aligned}\underline{apr}(A) &= \{x \in U \mid \mu_A(x) = 1\}\\ &= \text{core}(\mu_A),\\ \overline{apr}(A) &= \{x \in U \mid \mu_A(x) > 0\}\\ &= \text{support}(\mu_A).\end{aligned} \tag{16}$$

They are indeed the core and support of the fuzzy set μ_A.

The lower and upper approximations may be viewed as two operators from 2^U to 2^U, where 2^U is the power set of U. The system $(2^U, \neg, \cap, \cup, \underline{apr}, \overline{apr})$ is called a rough set algebra [13]. It may be considered as the Boolean algebra $(2^U, \neg, \cap, \cup)$ with two added unary operators $\underline{apr}$ and $\overline{apr}$. One can verify that the approximation operators satisfy the following conditions: for subsets $A,B\subseteq U$,

$$\begin{array}{ll}\text{(R1)} & \underline{apr}(A) = \neg(\overline{apr}(\neg A)),\\ & \overline{apr}(A) = \neg(\underline{apr}(\neg A)),\\ \text{(R2)} & \underline{apr}(A\cap B) = \underline{apr}(A) \cap \underline{apr}(B),\\ & \overline{apr}(A\cap B) \subseteq \overline{apr}(A) \cap \overline{apr}(B),\\ \text{(R3)} & \underline{apr}(A\cup B) \supseteq \underline{apr}(A) \cup \underline{apr}(B),\\ & \overline{apr}(A\cup B) = \overline{apr}(A) \cup \overline{apr}(B),\\ \text{(R4)} & \underline{apr}(A) \subseteq A \subseteq \overline{apr}(A),\end{array}$$

Property (R1) suggests that approximation operators are a pair of dual operators. By (R2) and (R3), the lower approximation operator is distributive over set intersection, and the upper approximation operators is distributive over set union. By comparing with (F1)-(F4), we can see that rough set approximation operators satisfy the same properties of core and support of fuzzy sets. In other words, fuzzy sets in general and fuzzy sets defined in rough set theory share the same qualitative characteristics.

Rough set approximations can be re-expressed in the following equivalent forms:

$$\begin{aligned}\underline{apr}(A) &= \{x \in U \mid [x]_E \subseteq A\} \\ &= \{x \in U \mid \forall y \in U(xEy \Longrightarrow y \in A)\}, \\ &= \bigcup\{[x]_E \mid [x]_E \subseteq A\}, \\ \overline{apr}(A) &= \{x \in U \mid [x]_E \cap A \neq \emptyset\} \\ &= \{x \in U \mid \exists y \in U(xEy, y \in A)\} \\ &= \bigcup\{[x]_E \mid [x]_E \cap A \neq \emptyset\}. \end{aligned} \tag{17}$$

They represent two additional definitions of rough set approximations [19]. The element based definition focuses on single elements of U. An element $x \in U$ is in the lower approximation, i.e., $\mu_A(x) = 1$, if and only if for all $y \in U$, xEy implies $y \in A$. An element $x \in U$ is in the upper approximation, i.e., $\mu_A(x) \neq 0$, if and only if there exists a $y \in U$ such that xEy and $y \in A$. This definition clearly relates rough set theory to modal logic [20]. It is useful in the generalization of rough sets using non-equivalence relations. On the other hand, the equivalence class (granule) based definition focuses on equivalence classes of the partition U/E. The lower approximation $\underline{apr}(A)$ is the union of those equivalence classes which are subsets of A. The upper approximation $\overline{apr}(A)$ is the union of those equivalence classes which have non-empty intersections with A. This definition is useful when generalizing rough sets using coverings of the universe. The equivalence of the two definitions can be seen from property (rm3), namely, elements in the same equivalence class have the same membership.

Rough membership functions describe a particular type of fuzziness induced by indiscernibility of elements. In an approximation space $apr = (U, E)$, a set is referred to as non-fuzzy if it generates a crisp subset of U. More specifically, a set is non-fuzzy if $\underline{apr}(A) = A = \overline{apr}(A)$. In other words, a subset A of U is non-fuzzy if the indiscernibility does not lead to fuzziness in our perception of A. The empty set, the equivalence classes, and unions of equivalence classes are non-fuzzy. The set of non-fuzzy sets, denoted by $\mathbf{Def}(U)$, consists of $\emptyset$ and U, and is closed under $\neg$, $\cap$ and $\cup$. That is, $\mathbf{Def}(U)$ is a sub-Boolean algebra of 2^U. All other subsets of U induce non-crisp fuzzy sets. By the equivalence class based definition, both $\mu_{\underline{apr}(A)}$ and $\mu_{\overline{apr}(A)}$ are crisp sets $\underline{apr}(A)$ and $\overline{apr}(A)$.

4.3 Main Features of the Rough Set Based Semantics of Fuzzy Sets

From the proposed semantic interpretation of fuzzy sets in the theory of rough sets, one can observe several important features. They are important for understanding the semantics of fuzzy sets. They are summarized below.

Fuzzy sets as a derived notion. Fuzzy sets are not treated as the primitive notion, but a derived notion. In particular, a fuzzy set is generated by a crisp set using an equivalence relation. By treating fuzzy sets as a derived notion, one is able to develop different semantics of fuzzy sets. Other primitive notions, which have sound semantic interpretations, can be used to define fuzzy sets. In rough set theory, the notion of equivalence relations can be easily defined and explained using an information table [8].

Representation of the source of fuzziness. The indiscernibility, represented by an equivalence relation, is the source of fuzziness in rough set theory. With respect to indiscernibility, some subsets of the universe cannot be precisely described, and are consequently described by fuzzy subsets. By explicitly defining the source and the form of fuzziness, we have a clear understanding of the meaning of fuzzy sets. Furthermore, even under the indiscernibility of elements, certain subsets of the universe are non-fuzzy. The distinction made between fuzzy and non-fuzzy subsets may provide more insights to the semantics of fuzzy sets.

Qualitative representation of fuzzy sets. The core and support are qualitative representations of a fuzzy set. In rough set theory, the core and support are referred to as the lower and upper approximations of a set. Although fuzzy sets defined in rough set theory are a special class of fuzzy sets, they share the same qualitative properties of any fuzzy set system defined by a pair of dual t-norm and t-conorm. This implies that the semantics of fuzzy sets presented here is not as restricted as it appears.

Constraints on fuzzy membership functions. The proposed semantics of fuzzy sets clearly shows two types of constraints on fuzzy membership functions. One type of constraints suggests that the membership values of two related elements are related. The other type of constraints suggests that fuzzy membership functions generated from two related sets (concepts) should be related. It should also be pointed out that the constraints on fuzzy membership functions are of a qualitative nature.

Constraints on fuzzy set-theoretic operators. Within rough set theory, there are constraints on fuzzy set-theoretic operators. It is clearly shown that fuzzy set-theoretic operators are not truth-functional.

Our discussions rely on a qualitative characterization of fuzzy sets. Rough set approximations are core and support of fuzzy sets. The t-norms and t-conorms deal with qualitative requirements on fuzzy set-theoretic operators. The qualitative semantic interpretation of fuzzy sets may provide more insights into the theory of fuzzy sets.

5 Generalized Rough Sets by Non-equivalence Relations

Standard rough sets are defined based on an equivalence relation, i.e., a reflexive, symmetric and transitive relation, on a finite and non-empty universe [7]. Yao *et al.* considered generalized rough sets based on an arbitrary binary relation [13, 20].

5.1 Binary Relations

Let $R \subseteq U \times U$ be a binary relation on a finite and non-empty universe U. For two elements $x, y \in U$, if xRy, we say that x is R-related to y. Let

$$\begin{aligned} R_s(x) &= \{y \in U \mid xRy\}, \\ R_p(x) &= \{y \in U \mid yRx\}, \end{aligned} \tag{18}$$

denote the successor and the predecessor neighborhoods of x induced by the binary relation R. Additional neighborhoods defined by a binary relation have also been studied [12, 17].

Different classes of binary relations can be obtained by considering the following properties:

$$\begin{array}{ll} \text{serial}: & \forall x \in U \exists y \in U[xRy], \\ & \forall x \in U[R_s(x) \neq \emptyset], \\ \text{inverse serial}: & \forall x \in U \exists y \in U[yRx], \\ & \forall x \in U[R_p(x) \neq \emptyset], \\ \text{reflexive}: & \forall x \in U[xRx], \\ & \forall x \in U[x \in R_s(x)], \\ \text{symmetric}: & \forall x, y \in U[xRy \Longrightarrow yRx], \\ & \forall x, y \in U[y \in R_s(x) \Longrightarrow x \in R_s(y)], \\ \text{transitive}: & \forall x, y, z \in U[(xRy, yRz) \Longrightarrow xRz], \\ & \forall x, y \in U[y \in R_s(x) \Longrightarrow R_s(y) \subseteq R_s(x)], \\ \text{Euclidean}: & \forall x, y, z \in U[(xRy, xRz) \Longrightarrow yRz], \\ & \forall x, y \in U[y \in R_s(x) \Longrightarrow R_s(x) \subseteq R_s(y)]. \end{array}$$

A relation is called a tolerance (compatibility) relation if it is reflexive and symmetric.

From a binary relation R, we can define four binary relations:

$$\begin{aligned} x \equiv_R y &\Longleftrightarrow R_s(x) = R_s(y), \\ x \approx_R y &\Longleftrightarrow R_s(x) \cap R_s(y) \neq \emptyset \\ x \simeq_R y &\Longleftrightarrow R_p(x) = R_p(y), \\ x \sim_R y &\Longleftrightarrow R_p(x) \cap R_p(y) \neq \emptyset. \end{aligned} \tag{19}$$

Relations $\equiv_R$ and $\simeq_R$ are reflexive, symmetric, and transitive, and hence are equivalence relations. Relations $\approx_R$ and $\sim_R$ are symmetric. If R is a serial relation, then $\approx_R$ is also reflexive, and hence a tolerance relation. If R is inverse serial, $\sim_R$ is reflexive, and hence a tolerance relation. While $\equiv_R$ and $\simeq_R$ show strong connections between elements of U, $\approx_R$ and $\sim_R$ show some weak connections. Two elements are considered to be equivalent if they have the same successor (predecessor) neighborhood, and similar if their successor (predecessor) neighborhoods have a non-empty overlap.

5.2 Rough Membership Functions

The pair $apr = (U, R)$ is called an approximation space, with the relation R representing the relationships between elements of U. One can define rough membership functions by extending equation (14) using neighborhoods induced by a binary relation [14]. We consider the definition based on the successor neighborhoods. For a subset $A \subseteq U$, a rough membership function can be defined by substituting $[x]_E$ with $R_s(x)$ in equation (14) as follows:

$$\mu_A(x) = \frac{|R_s(x) \cap A|}{|R_s(x)|}. \tag{20}$$

We assume that relation R is at least serial, i.e., $R_s(x) \neq \emptyset$ for all $x \in U$. Under this assumption, the rough membership function is well defined.

Recall that we can identify at least two types of constraints. One type suggests that membership values of related elements should be related. Another type suggests that two rough membership functions obtained from two related generating sets should be related. It is naturally expected that relationships given by binary relations should impose some constraints on rough membership functions.

For an arbitrary binary relation R, we have:

$$\text{(I)} \qquad xRy \Longleftrightarrow \mu_{\{y\}}(x) \neq 0.$$

If the relation R is reflexive, the following condition holds:

$$\text{(II)} \qquad \mu_{R_s(x)}(x) = 1.$$

They serve as basic properties for us to derive other properties of rough membership functions. Suppose the relation R is both serial and inverse serial. The rough membership functions still satisfy (rm1), (rm2), and (rm6). Corresponding to property (rm3), we have the following constraints:

$$\begin{array}{ll}
\text{(rm3.1)} & x \equiv_R y \Longrightarrow \mu_A(x) = \mu_A(y); \\
\text{(rm3.2)} & (\mu_A(x) = 1, x \approx_R y) \Longrightarrow \mu_A(y) \neq 0, \\
 & (\mu_A(x) = 0, x \approx_R y) \Longrightarrow \mu_A(y) \neq 1, \\
\text{(rm3.3)} & x \simeq_R y \Longrightarrow \mu_{\{x\}}(z) = \mu_{\{y\}}(z), \\
\text{(rm3.4)} & (\mu_{\{x\}}(z) \neq 0, \neg(x \sim_R y)) \Longrightarrow \mu_{\{y\}}(z) = 0.
\end{array}$$

The first two properties state one type of constraints, namely, membership values of related elements are related. According to (rm3.1), if two elements are equivalent, in the sense defined by $\equiv_R$, then they must have the same membership value. According to (rm3.2), if an element y is similar to another element x with full membership, in the sense defined by $\approx_R$, then y cannot have a null membership. Likewise, if y is similar to an element x with null membership, then y cannot have a full membership. The last two properties show another type of constraints, namely, rough membership functions derived from two related generating sets must be related. By (rm3.3), if x and y are equivalent, in the sense of $\simeq_R$, we can conclude that $\{x\}$ and $\{y\}$ generate the same rough membership function. By (rm3.4), if z has a non-zero membership value in $\mu_{\{x\}}$ and $\neg(x \sim_R y)$, we can conclude that z must have a zero membership value in

$\mu_{\{y\}}$. That is, if x is not similar to y, x and y generate totally different rough membership functions in the sense that whenever z has a non-zero membership value for one, it must have the zero membership value for the other.

Rough membership values defined using an arbitrary binary relation are also conditional probabilities. Consequently, properties (o1)-(o4) hold for rough set operators.

5.3 Rough Set Approximation and Approximation Operators

Similar to the classical case, a pair of rough set approximations can be defined by the core and support of μ_A. That is, for a set $A \subseteq U$, we have [13, 20]:

$$\begin{aligned}
\underline{apr}(A) &= \mathrm{core}(\mu_A), \\
&= \{x \in U \mid \mu_A(x) = 1\} \\
&= \{x \in U \mid R_s(x) \subseteq A\} \\
&= \{x \in U \mid \forall y \in U(xRy \Longrightarrow y \in A)\}, \\
\overline{apr}(A) &= \mathrm{support}(\mu_A) \\
&= \{x \in U \mid \mu_A(x) > 0\} \\
&= \{x \in U \mid R_s(x) \cap A \neq \emptyset\} \\
&= \{x \in U \mid \exists y \in U(xRy, y \in A)\}. \qquad (21)
\end{aligned}$$

Independent of the properties of the binary relation, they satisfy properties (R1)-(R3).

Properties of a binary relation determine the properties of approximation operators $\underline{apr}$ and $\overline{apr}$. With respect to the serial, reflexive, symmetric, transitive and Euclidean properties, the approximation operators have the corresponding properties:

$$\begin{aligned}
&(\mathrm{D}) && \underline{apr}(A) \subseteq \overline{apr}(A); \\
&(\mathrm{T}) && \underline{apr}(A) \subseteq A, \\
& && A \subseteq \overline{apr}(A); \\
&(\mathrm{B}) && A \subseteq \underline{apr}(\overline{apr}(A)), \\
& && \overline{apr}(\underline{apr}(A)) \subseteq A; \\
&(4) && \underline{apr}(A) \subseteq \underline{apr}(\underline{apr}(A)), \\
& && \overline{apr}(\overline{apr}(A)) \subseteq \overline{apr}(A); \\
&(5) && \overline{apr}(A) \subseteq \underline{apr}(\overline{apr}(A)), \\
& && \overline{apr}(\underline{apr}(A)) \subseteq \underline{apr}(A).
\end{aligned}$$

By combining properties of binary relations, we can produce more classes of rough set models.

5.4 Classes of Rough Set Models

With respect to properties of a binary relation, such as reflexive, symmetric, transitive, and Euclidean, we can identify additional constraints on rough membership functions. For simplicity, we only consider constraints on membership values with respect to the same fuzzy set. In this case, they are the re-expression of properties (T), (B), (4) and (5) in terms of rough membership functions. Thus, we use the same symbols to label the corresponding properties.

Reflexive Model. A reflexive relation is a special serial relation. In this case, for all $x \in U$, xRx or equivalently $x \in R_s(x)$. The corresponding equivalent properties of rough membership functions can be stated as:

$$\begin{array}{ll} \text{(T)} & \mu_{\{x\}}(x) \neq 0; \\ & x \in A \Longrightarrow \mu_A(x) \neq 0; \\ & \mu_A(x) = 1 \Longrightarrow x \in A. \end{array}$$

They are in fact the properties (rm4) and (rm5). The equivalence of the properties can be easily shown. As an example, we show the equivalence of the first and the last properties. Assume $\mu_{\{x\}}(x) \neq 0$. By (I), we can conclude that $x \in R_s(x)$. If $\mu_A(x) = 1$, we have $R_s(x) \subseteq A$. By combining $x \in R_s(x)$ and $R_s(x) \subseteq A$, we have $x \in A$. This means that the first property implies the last property. Now assume $\mu_A(x) = 1 \Longrightarrow x \in A$. According to (II), we have $\mu_{R_s(x)}(x) = 1$. Thus, $x \in R_s(x)$, which implies $\mu_{\{x\}}(x) \neq 0$. We have therefore showed that the last property implies the first property.

Symmetric Model. For a symmetric relation, if $x \in R_s(y)$ then $y \in R_s(x)$, namely, $R_p(x) = R_s(x)$. Relations $\equiv_R$ and $\simeq_R$ become the same, and relations $\approx_R$ and $\sim_R$ become the same. In a symmetric model, we have the following equivalent properties on a rough membership function:

$$\begin{array}{ll} \text{(B)} & \mu_{\{y\}}(x) \neq 0 \Longrightarrow \mu_{\{x\}}(y) \neq 0; \\ & [x \in A, xRy] \Longrightarrow \mu_A(y) \neq 0; \\ & [\mu_A(y) = 0, xRy] \Longrightarrow x \notin A; \\ & [\mu_A(y) = 1, xRy] \Longrightarrow x \in A; \\ & [x \notin A, xRy] \Longrightarrow \mu_A(y) \neq 1. \end{array}$$

These properties are similar, in form, to properties (rm3.2) and (rm3.4). Furthermore, they also contain conditions in (T). Property (B) states constraints on the membership functions directly through the binary relation R. For example, if y is R-related to an element $x \in A$, then y has a non-zero membership value in μ_A. On the other hand, (rm3.2) and (rm3.4) state constraints indirectly through R.

Transitive Model. For a transitive relation R, for all $x, y \in U$, if xRy, then $R_s(y) \subseteq R_s(x)$. For a transitive relation R, we have the connection between R and $\approx_R$ and $\sim_R$:

$$\begin{aligned} &R \text{ is serial and transitive } \Longrightarrow [xRy \Longrightarrow x \approx_R y]; \\ &R \text{ is inverse serial and transitive } \Longrightarrow [xRy \Longrightarrow x \sim_R y]. \end{aligned} \tag{22}$$

In a transitive model, we have the following equivalent properties on membership functions:

$$\begin{array}{ll}(4) & \mu_{\{y\}}(x) \neq 0 \Longrightarrow \mu_{R_s(x)}(y) = 1; \\ & [\mu_A(x) = 1, xRy] \Longrightarrow \mu_A(y) = 1; \\ & [\mu_A(y) \neq 1, xRy] \Longrightarrow \mu_A(x) \neq 1; \\ & [\mu_A(x) \neq 0, yRx] \Longrightarrow \mu_A(y) \neq 0; \\ & [\mu_A(y) = 0, yRx] \Longrightarrow \mu_A(x) = 0.\end{array}$$

Combining the above results with (rm3.2) and (rm3.4), we can obtain additional properties.

Euclidean Model. For an Euclidean relation, for all $x, y \in U$, if xRy, then $R_s(x) \subseteq R_s(y)$. In this case, we have:

$$\begin{aligned}&R \text{ is Euclidean } \Longrightarrow [xRy \Longrightarrow x \approx_R y];\\ &R \text{ is inverse serial and Euclidean } \Longrightarrow [xRy \Longrightarrow x \sim_R y]. \qquad (23)\end{aligned}$$

The following properties can be obtained:

$$\begin{array}{ll}(5) & \mu_{\{y\}}(x) \neq 0 \Longrightarrow \mu_{R_s(y)}(x) = 1; \\ & [\mu_A(x) \neq 0, xRy] \Longrightarrow \mu_A(y) \neq 0; \\ & [\mu_A(y) = 0, xRy] \Longrightarrow \mu_A(x) = 0; \\ & [\mu_A(x) = 1, yRx] \Longrightarrow \mu_A(y) = 1; \\ & [\mu_A(y) \neq 1, yRx] \Longrightarrow \mu_A(x) \neq 1.\end{array}$$

They can be used to infer additional properties.

Pawlak Model. A Pawlak model is characterized by an equivalence relation. In this case, all four relations $\equiv_R$, $\approx_R$, $\simeq_R$ and $\sim_R$ are the same as the relation R. An equivalence relation is an Euclidean relation. Thus, all the properties established so far hold for a Pawlak model.

6 Generalized Rough Sets Based on Coverings of the Universe

In a partition, an element belongs to one equivalence class and two distinct equivalence classes have no overlap. Covering of the universe is a natural generalization of partitions, in which an element can be in more than one class of a covering. This provides a more realistic model for the formulation of rough sets [14, 24].

6.1 Coverings

A covering of the universe, $\mathbf{C} = \{C_1, \ldots, C_n\}$, is a family of subsets of U such that $U = \bigcup\{C_i \mid i = 1, \ldots, n\}$. Two distinct sets in $\mathbf{C}$ may have a non-empty overlap. An element may belong to more than one class in $\mathbf{C}$. The family $\mathbf{C}(x) = \{C \in \mathbf{C} \mid x \in C\}$ consists of sets in $\mathbf{C}$ containing x. With respect to a covering, we construct the following equivalence relation:

$$\begin{aligned} x \equiv_{\mathbf{C}} y &\Longleftrightarrow \forall C \in \mathbf{C}(x \in C \Longleftrightarrow y \in C) \\ &\Longleftrightarrow \mathbf{C}(x) = \mathbf{C}(y). \end{aligned} \tag{24}$$

Two elements are considered to be equivalent if they appear in the same family of subsets in $\mathbf{C}$. A tolerance relation $\sim_{\mathbf{C}}$ can be defined by:

$$\begin{aligned} x \sim_{\mathbf{C}} y &\Longleftrightarrow \exists C \in \mathbf{C}(x \in C, y \in C) \\ &\Longleftrightarrow \mathbf{C}(x) \cap \mathbf{C}(y) \neq \emptyset. \end{aligned} \tag{25}$$

That is, x and y are considered to be similar if they appear together in at least one class of the covering.

6.2 Rough Membership Functions

The sets in $\mathbf{C}(x)$ describe different types or various degrees of similarity between elements of U. For a set $C \in \mathbf{C}(x)$, we may compute a value $|C \cap A|/|C|$ by extending equation (14). It may be interpreted as the membership value of x from the view point of C. With respect to a covering $\mathbf{C}$, we have a family of values $\{|C \cap A|/|C| \mid x \in C, C \in \mathbf{C}\}$. Generalized rough membership functions may be defined by using this family of values. We consider the following three definitions [22]:

$$\text{(minimum)} \qquad \underline{\mu}_A(x) = \min\left\{\frac{|C \cap A|}{|C|} \mid x \in C, C \in \mathbf{C}\right\}, \tag{26}$$

$$\text{(maximum)} \qquad \overline{\mu}_A(x) = \max\left\{\frac{|C \cap A|}{|C|} \mid x \in C, C \in \mathbf{C}\right\}, \tag{27}$$

$$\text{(average)} \qquad \mu^*_A(x) = \text{avg}\left\{\frac{|C_i \cap A|}{|C_i|} \mid x \in C, C \in \mathbf{C}\right\}. \tag{28}$$

The minimum, maximum, and average definitions may be regarded as the most permissive, the most optimistic, and a balanced view, respectively, in defining rough membership functions. The minimum rough membership function is determined by a set in $\mathbf{C}(x)$ which has the smallest overlap with A, and the maximum rough membership function by a set in $\mathbf{C}(x)$ which has the largest overlap with A. The average rough membership function depends on every set in $\mathbf{C}(x)$.

The generalized rough membership functions have the following properties:

(grm0) $\underline{\mu}_A(x) \leq \mu^*_A(x) \leq \overline{\mu}_A(x)$,
(grm1) $\underline{\mu}_U(x) = \mu^*_U(x) = \overline{\mu}_U(x) = 1$,
(grm2) $\underline{\mu}_\emptyset(x) = \mu^*_\emptyset(x) = \overline{\mu}_\emptyset(x) = 0$,

$$\begin{array}{ll}
\text{(grm3)} & x \equiv_{\mathbf{C}} y \Longrightarrow [\underline{\mu}_A(x) = \underline{\mu}_A(y), \mu^*_A(x) = \mu^*_A(y), \overline{\mu}_A(x) = \overline{\mu}_A(y)], \\
\text{(grm4)} & (x \sim_{\mathbf{C}} y, \underline{\mu}_A(x) = 1) \Longrightarrow \overline{\mu}_A(y) = 1, \\
 & (x \sim_{\mathbf{C}} y, \overline{\mu}_A(x) = 0) \Longrightarrow \underline{\mu}_A(y) = 0, \\
 & (x \sim_{\mathbf{C}} y, \overline{\mu}_A(x) = 0) \Longrightarrow \mu^*_A(y) \neq 1, \\
 & (x \sim_{\mathbf{C}} y, \underline{\mu}_A(x) = 1) \Longrightarrow \mu^*_A(y) \neq 0, \\
\text{(grm5)} & x \in A \Longrightarrow \underline{\mu}_A(x) \neq 0, \\
 & \underline{\mu}_A(x) = 0 \Longrightarrow x \notin A, \\
 & x \in A \Longrightarrow \overline{\mu}_A(x) \neq 0, \\
 & \overline{\mu}_A(x) = 0 \Longrightarrow x \notin A, \\
\text{(grm6)} & \overline{\mu}_A(x) = 1 \Longrightarrow x \in A, \\
 & x \notin A \Longrightarrow \overline{\mu}_A(x) \neq 1, \\
 & \underline{\mu}_A(x) = 1 \Longrightarrow x \in A, \\
 & x \notin A \Longrightarrow \underline{\mu}_A(x) \neq 1, \\
\text{(grm7)} & A \subseteq B \Longrightarrow \underline{\mu}_A(x) \leq \underline{\mu}_B(x), \\
 & A \subseteq B \Longrightarrow \overline{\mu}_A(x) \leq \overline{\mu}_B(x).
\end{array}$$

Property (grm0) states the relationship between three membership functions. A partition is a special type of covering. In this case, three rough membership functions reduce to the same rough membership function. Properties (grm1) and (grm2) show the membership values of two special sets, the empty set $\emptyset$ and the entire set U. Both (grm3) and (grm4) show the constraints on rough membership functions imposed by the similarity of objects. From the relation $\underline{\mu}_A(x) \leq \mu^*_A(x) \leq \overline{\mu}_A(x)$, we can obtain additional properties. For example, (grm5) implies that $x \in A \Longrightarrow \mu^*_A(x) \neq 0$. Similarly, (grm6) implies that $\mu^*_A(x) = 1 \Longrightarrow x \in A$.

For set-theoretic operators, one can verify the following properties:

$$\begin{aligned}
&\underline{\mu}_{\neg A}(x) = 1 - \overline{\mu}_A(x), \\
&\overline{\mu}_{\neg A}(x) = 1 - \underline{\mu}_A(x), \\
&\mu^*_{\neg A}(x) = 1 - \mu^*_A(x), \\
&\max(0, \underline{\mu}_A(x) + \underline{\mu}_B(x) - \overline{\mu}_{A\cup B}(x)) \leq \underline{\mu}_{A\cap B}(x) \leq \min(\underline{\mu}_A(x), \underline{\mu}_B(x)), \\
&\max(\overline{\mu}_A(x), \overline{\mu}_B(x)) \leq \overline{\mu}_{A\cup B}(x) \leq \min(1, \overline{\mu}_A(x) + \overline{\mu}_B(x) - \underline{\mu}_{A\cap B}(x)), \\
&\mu^*_{A\cap B}(x) = \mu^*_A(x) + \mu^*_B(x) - \mu^*_{A\cup B}(x).
\end{aligned} \tag{29}$$

We again obtain non-truth-functional rough set operators.

The minimum rough membership function may be viewed as the lower bound on all possible rough membership functions definable using a covering, while the maximum rough membership as the upper bound. The pair $(\underline{\mu}_A(x), \overline{\mu}_A(x))$ may also be used to define an interval-valued fuzzy set [6]. The interval $[\underline{\mu}_A(x), \overline{\mu}_A(x)]$ is the membership value of x with respect to A.

6.3 Rough Set Approximations and Approximation Operators

From the three rough membership functions, we define three pairs of lower and upper approximations. For the minimum definition, we have:

$$\begin{aligned}
\underline{apr}^{m}(A) &= \mathrm{core}(\underline{\mu}_A) \\
&= \{x \in U \mid \underline{\mu}_A(x) = 1\} \\
&= \{x \in U \mid \forall C \in \mathbf{C}(x \in C \Longrightarrow C \subseteq A)\}, \\
\overline{apr}^{m}(A) &= \mathrm{support}(\underline{\mu}_A) \\
&= \{x \in U \mid \underline{\mu}_A(x) > 0\} \\
&= \{x \in U \mid \forall C \in \mathbf{C}(x \in C \Longrightarrow C \cap A \neq \emptyset)\}.
\end{aligned} \tag{30}$$

For the maximum definition, we have:

$$\begin{aligned}
\underline{apr}^{M}(A) &= \mathrm{core}(\overline{\mu}_A) \\
&= \{x \in U \mid \overline{\mu}_A(x) = 1\} \\
&= \{x \in U \mid \exists C \in \mathbf{C}(x \in C, C \subseteq A)\}, \\
&= \bigcup \{C \in \mathbf{C} \mid C \subseteq A\}, \\
\overline{apr}^{M}(A) &= \mathrm{support}(\overline{\mu}_A) \\
&= \{x \in U \mid \overline{\mu}_A(x) > 0\} \\
&= \{x \in U \mid \exists C \in \mathbf{C}(x \in C, C \cap A \neq \emptyset)\} \\
&= \bigcup \{C \in \mathbf{C} \mid C \cap A \neq \emptyset\}.
\end{aligned} \tag{31}$$

The lower and upper approximations in each pair are no longer dual operators. However, $(\underline{apr}^{m}, \overline{apr}^{M})$ and $(\underline{apr}^{M}, \overline{apr}^{m})$ are two pairs of dual operators. The first pair can be derived from the average definition, namely:

$$\underline{apr}^{*}(A) = \underline{apr}^{m}(A), \qquad \overline{apr}^{*}(A) = \overline{apr}^{M}(A). \tag{32}$$

These approximation operators have been studied extensively in rough set theory. Their connections and properties can be found in a recent paper by Yao [17].

7 Conclusion

By applying the techniques used in the study of rough sets, we propose a semantic framework for fuzzy sets. Two salient features of the proposed framework are the consideration of knowledge about the universe and the incorporation of a method for constructing fuzzy sets. This leads to a constructive fuzzy set theory, where fuzzy sets are explicitly constructed and interpreted.

Our examination of fuzzy sets and rough sets suggests that both theories share the same qualitative properties. The lower and upper approximations of rough set theory correspond to the core and support of fuzzy set theory. Several methods are suggested for the construction of fuzzy sets, depending on the available knowledge. Fuzzy membership functions are defined and interpreted in classical and generalized rough set models.

The connections established between the two theories may enhance our understanding of each theory. The results may provide a possible solution to a fundamental difficulty with fuzzy set theory regarding the semantic interpretations of fuzzy membership values.

References

1. Dubois, D. and Prade, P. A class of fuzzy measures based on triangular norms, *International Journal of General Systems*, **8**, 43-61, 1982.
2. Dubois, D. and Prade, H. Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems*, **17**, 191-209, 1990.
3. Dubois, D. and Prade, P. Similarity-based approximate reasoning, in: J.M. Zurada, R.J. Marks II, C.J. Robinson (Eds.), *Computational Intelligence: Imitating Life*, IEEE Press, New York, 69-80, 1994.
4. Dubois, D. and Prade, P. The three semantics of fuzzy set, *Fuzzy Sets and Systems*, **90**, 141-150, 1997.
5. Klement, E.P., Mesiar, R. and Pap, E. *Triangular Norms*, Kluwer Academic Publishers, Dordrecht, 2000.
6. Klir, G.J. and Yuan, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, New Jersey, 1995.
7. Pawlak, Z. Rough sets, *International Journal of Computer and Information Sciences*, **11**, 341-356, 1982.
8. Pawlak, Z. *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.
9. Pawlak, Z. and Skowron, A. Rough membership functions, in: *Advances in the Dempster-Shafer Theory of Evidence*, R.R. Yager and M. Fedrizzi and J. Kacprzyk (Eds.), John Wiley and Sons, New York, 251-271, 1994.
10. Radzikowska, A.M. and Kerre, E.E. Fuzzy rough sets based on residuated lattices, in: *Rough, Fuzzy and Fuzzy Rough Sets*, D. Dubois, J. Grzymala-Busse, M. Inuiguchi, and L. Polkowski, Springer, Berlin, 2004.
11. Wong, S.K.M. and Ziarko, W. Comparison of the probabilistic approximate classification and the fuzzy set model, *Fuzzy Sets and Systems*, **21**, 357-362, 1987.
12. Wu, W.Z. and Zhang, W.X. Neighborhood operator systems and approximations, *Information Sciences*, **144**, 201-217, 2002.
13. Yao, Y.Y. Two views of the theory of rough sets in finite universes, *International Journal of Approximation Reasoning*, **15**, 291-317, 1996.
14. Yao, Y.Y. A comparative study of fuzzy sets and rough sets, *Information Sciences*, **109**, 227-242, 1998.
15. Yao, Y.Y. Generalized rough set models, in: *Rough Sets in Knowledge Discovery*, Polkowski, L. and Skowron, A. (Eds.), Physica-Verlag, Heidelberg, 286-318, 1998.
16. Yao, Y.Y. Constructive and algebraic methods of the theory of rough sets, *Information Sciences*, **109**, 21-47, 1998.
17. Yao, Y.Y. Relational interpretations of neighborhood operators and rough set approximation operators, *Information Sciences*, **111**, 239-259, 1998.
18. Yao, Y.Y. Rough sets and interval fuzzy sets, *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2347-2352, 2001.

19. Yao, Y.Y. Information granulation and approximation in a decision-theoretic model of rough sets, manuscript, 2002.
20. Yao, Y.Y. and Lin, T.Y. Generalization of rough sets using modal logic, *Intelligent Automation and Soft Computing, an International Journal*, **2**, 103-120, 1996.
21. Yao, Y.Y. and Wong, S.K.M. A decision theoretic framework for approximating concepts, *International Journal of Man-machine Studies*, **37**, 793-809, 1992.
22. Yao, Y.Y. and Zhang, J.P. Interpreting fuzzy membership functions in the theory of rough sets, *Proceedings of The Second International Conference on Rough Sets and Current Trends in Computing*, LNAI 2005, 82-89, 2001.
23. Zadeh, L.A. Fuzzy sets, *Information and Control*, **8**, 338-353, 1965.
24. Zakowski, W. Approximations in the space (U, Π), *Demonstratio Mathematica*, *XVI*, 761-769, 1983.
25. Zhang, B. and Zhang, L. *Theory and Applications of Problem Solving*, North-Holland, Amsterdam, 1992.

A New Proposal for Fuzzy Rough Approximations and Gradual Decision Rule Representation

Salvatore Greco[1], Masahiro Inuiguchi[2], and Roman Słowiński[3,4]

[1] Faculty of Economics, University of Catania, Corso Italia, 55,
95129 Catania, Italy
salgreco@mbox.unict.it

[2] Department of Systems Innovation Graduate School of Engineering Science,
Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
inuiguti@sys.es.osaka-u.ac.jp

[3] Institute of Computing Science, Poznan University of Technology,
60-965 Poznan, Poland
slowinsk@sol.put.poznan.pl

[4] Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. We propose a new fuzzy rough set approach which, differently from most of known fuzzy set extensions of rough set theory, does not use any fuzzy logical connectives (t-norm, t-conorm, fuzzy implication). As there is no rationale for a particular choice of these connectives, avoiding this choice permits to reduce the part of arbitrary in the fuzzy rough approximation. Another advantage of the new approach is that it uses only the ordinal property of fuzzy membership degrees. The concepts of fuzzy lower and upper approximations are thus proposed, creating a base for induction of fuzzy decision rules having syntax and semantics of gradual rules. The decision rules are induced from lower and upper approximations defined for positive and negative relationships between credibility of premise and conclusion; for this reason, there are four types of decision rules. In addition to decision rule representation, a new scheme of inference with a generalized fuzzy rough *modus ponens* is proposed.

Keywords: rough sets, fuzzy sets, decision rules, gradual rules, ambiguity, credibility, *modus ponens*

1 Introduction

It has been acknowledged by different studies that fuzzy set theory and rough set theory are complementary because of handling different kinds of uncertainty. Fuzzy sets deal with possibilistic uncertainty, connected with imprecision of states, perceptions and preferences (Dubois, Prade and Yager [8]). Rough sets deal, in turn, with uncertainty following from ambiguity of information (Pawlak [21]). The two types of uncertainty can be encountered together in real-life problems. For this reason, many approaches have been proposed to combine

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 319–342, 2004.

fuzzy sets with rough sets (Cattaneo [5], Dubois, Prade [7], Greco, Matarazzo, Słowiński [11–13], Inuiguchi, Tanino [18], Nakamura [19], Nakamura and Gao [20], Polkowski [23], Słowiński [24], Słowiński, Stefanowski [25], Yao [27]). The main preoccupation in almost all these studies was related to fuzzy extension of Pawlak's definition of lower and upper approximation using fuzzy connectives (t-norm, t-conorm, fuzzy implication). In fact, there is no rule for the choice of the "right" connective, so this choice is always arbitrary to some extend.

Another drawback of fuzzy extensions of rough sets involving fuzzy connectives is that they are based on cardinal properties of membership degrees. In consequence, the result of these extensions is sensitive to order preserving transformation of membership degrees. For example, consider the t-conorm of Łukasiewicz as a fuzzy connective; it may be used in the definition of both fuzzy lower approximation (to build fuzzy implication) and fuzzy upper approximation (as a fuzzy counterpart of a union). The t-conorm of Łukasiewicz is defined as,

$$T^*(\alpha,\beta) = \min(\alpha+\beta,1).$$

Let us consider the following values of arguments:

$$\alpha = 0.5,\ \beta = 0.3,\ \gamma = 0.2,\ \delta = 0.1,$$

and their order preserving transformation:

$$\alpha' = 0.4,\ \beta' = 0.3,\ \gamma' = 0.2,\ \delta' = 0.05.$$

The values of the t-conorm are in both cases as follows:

$$\begin{array}{ll} T^*(\alpha,\delta) = 0.6, & T^*(\beta,\gamma) = 0.5, \\ T^*(\alpha',\delta') = 0.45, & T^*(\beta',\gamma') = 0.5. \end{array}$$

One can see that the order of the results has changed after the order preserving transformation of the arguments. This means that the Łukasiewicz t-conorm takes into account not only the ordinal properties of the membership degrees, but also their cardinal properties. A natural question arises: is it reasonable to expect from membership degree a cardinal content instead of ordinal only? Or, in other words, is it realistic that human is able to say in a meaningful way not only that

a) "object x belongs to fuzzy set X more likely than object y",

but even something like

b) "object x belongs to fuzzy set X two times more likely than object y"?

We claim that it is safer to consider information of type a), because information of type b) is rather meaningless for humans (see Bilgic, Turksen [3]).

The above doubt about the cardinal content of the fuzzy membership degree shows the need for methodologies which consider the imprecision in perception typical for fuzzy sets but avoid as much as possible meaningless transformation of information through fuzzy connectives.

The approach we propose for fuzzy extension of rough sets takes into account the above request. It avoids arbitrary choice of fuzzy connectives and not meaningful operations on membership degrees. Our approach belongs to the minority of fuzzy extensions of the rough set concept that do not involve fuzzy connectives and cardinal interpretation of membership degrees. Within this minority, it is related to the approach of Nakamura and Gao [20] using α-cuts on fuzzy similarity relation between objects; this relationship will be explained in the next section.

We propose a methodology of fuzzy rough approximation that infers the most cautious conclusion from available imprecise information. In particular, we observe that any approximation of knowledge about Y using knowledge about X is based on positive or negative relationships between premise and conclusion, i.e,

i) "the more x is X, the more it is Y" (positive relationship),
ii) "the more x is X, the less it is Y" (negative relationship).

The following simple relationships illustrate i) and ii): "the larger the market share of a company, the larger its profit" (positive relationship) and "the larger the debt of a company, the smaller its profit" (negative relationship). These relationships have been already considered within fuzzy set theory under the name of *gradual rules* (Dubois, Prade [6]). In this article, we are extending the concept of gradual rules by handling ambiguity of information through fuzzy rough approximations.

Remark that the syntax of gradual rules is based on monotonic relationship that can also be found in dominance-based decision rules induced from preference-ordered data. From this point of view, the fuzzy rough approximation proposed in this article is related to the dominance-based rough set approach (Greco, Matarazzo, Słowiński [11, 14, 15]).

For the reason of greater generality, one could eventually drop the assumption of the monotonic relationship between premise and conclusion in gradual rules. For example, the gradual rule "the greater the temperature the better the weather" is true in some range of temperature only (say, up to 30 degrees C). In such cases, however, one can split the domain of the premise into sub-intervals, in which monotonicity still holds, and represent the regularities observed in these sub-intervals by gradual rules. For example, we can split the range of the temperature into two open subintervals, under 30 degrees C and over 30 degrees C, obtaining the two gradual rules: "the greater the temperature the better the weather", which is valid in the first interval, and "the smaller the temperature the better the weather", which is valid in the second interval. Therefore, the concept of monotonicity in gradual rules is intrinsic to the idea of induction whose aim is to represent regularities according to the simplest law (see, Proposition 6.363 of Wittgenstein [26]: "*The process of induction is the process of assuming the simplest law that can be made to harmonize with our experience*"). We claim that this simplest law is monotonicity.

The above Proposition of Wittgenstein [26] is borrowed from the paper by Aragones, Gilboa, Postelwaite, and Schmeidler [1] on a similar subject. Remark, however, that these authors consider rules with non-monotonic relationships between premise and conclusion, and, moreover, their rule induction procedure is based on a cardinal concept of the credibility of information.

The plan of the article is the following. In the next section, we are defining a syntax and a semantics of considered gradual rules; we also show how they represent positive and negative relationships between fuzzy sets corresponding to premise and conclusion of a decision rule. In Sect. 3, we are introducing fuzzy rough approximations consistent with the considered gradual rules; the fuzzy rough approximations create a base for induction of decision rules. In Sect. 4, we propose a new scheme of inference with a fuzzy rough *modus ponens* adapted to the considered decision rules. Sect. 5 is grouping conclusions and remarks on further research directions.

2 Decision Rules with Positively or Negatively Related Premise and Conclusion

We aim to obtain gradual decision rules of the following types,

- *lower-approximation rule with positive relationship* (LP-rule): "if condition X has credibility $C(X) \geq \alpha$, then decision Y has credibility $C(Y) \geq \beta$";
- *lower-approximation rule with negative relationship* (LN-rule): "if condition X has credibility $C(X) \leq \alpha$, then decision Y has credibility $C(Y) \geq \beta$";
- *upper-approximation rule with positive relationship* (UP-rule): "if condition X has credibility $C(X) \leq \alpha$, then decision Y could have credibility $C(Y) \leq \beta$";
- *upper-approximation rule with negative relationship* (UN-rule): "if condition X has credibility $C(X) \geq \alpha$, then decision Y could have credibility $C(Y) \leq \beta$".

The above decision rules will be represented by triples $\langle X, Y, f^+ \rangle$, $\langle X, Y, f^- \rangle$, $\langle X, Y, g^+ \rangle$ and $\langle X, Y, g^- \rangle$, respectively, where X is a given condition, Y is a given decision and $f^+ : [0,1] \rightarrow [0,1]$, $f^- : [0,1] \rightarrow [0,1]$, $g^+ : [0,1] \rightarrow [0,1]$ and $g^- : [0,1] \rightarrow [0,1]$ are functions relating credibility of X to credibility of Y in lower- and upper-approximation rules, respectively. More precisely, functions f^+, f^-, g^+ and g^- permit to rewrite the above decision rules as follows,

- LP-rule: "if condition X has credibility $C(X) \geq \alpha$, then decision Y has credibility $C(Y) \geq \beta = f^+(\alpha)$";
- LN-rule: "if condition X has credibility $C(X) \leq \alpha$, then decision Y has credibility $C(Y) \geq \beta = f^-(\alpha)$";
- UP-rule: "if condition X has credibility $C(X) \leq \alpha$, then decision Y could have credibility $C(Y) \leq \beta = g^+(\alpha)$".
- UN-rule: "if condition X has credibility $C(X) \geq \alpha$, then decision Y could have credibility $C(Y) \leq \beta = g^-(\alpha)$".

An LP-rule can be regarded as a gradual rule (Dubois, Prade [6]); it can be interpreted as: "the more object x is X, the more it is Y". In this case, the relationship between credibility of premise and conclusion is positive and certain. LN-rule can be interpreted in turn as: "the less object x is X, the more it is Y", so the relationship is negative and certain. On the other hand, the UP-rule can be interpreted as: "the more object x is X, the more it could be Y", so the relationship is positive and possible. Finally, UN-rule can be interpreted as: "the less object x is X, the more it could be Y", so the relationship is negative and possible.

Example A. In this example we shall illustrate the usefulness of the above definitions of four types of decision rules.

Let us consider a hypothetical car selection problem in which the maximum speed is used for evaluation of cars.

We assume that the decision maker evaluates 10 sample cars, listed in Table 1, from the viewpoint of acceptability for his/her purchase. The evaluation score is given as a degree of acceptability between 0 (strong rejection) and 1 (strong acceptance), as shown in the bottom row of Table 1.

Table 1. Decision maker's evaluation of sample cars.

Car	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
maximum_speed (km/h)	180	200	210	200	190	240	210	220	200	230
Acceptability	0.3	0.4	0.6	0.8	0.2	0.9	0.7	0.9	0.7	1

In the decision maker's opinion, a car whose *maximum_speed* is more than 230 km/h is definitely speedy and a car whose *maximum_speed* is less than 190 km/h is definitely not speedy. Therefore, we may define fuzzy set *speedy_car* for the decision maker by the following membership function,

$$\mu_{speedy\_car}(maximum\_speed(x)) = \begin{cases} 0 & \text{if } maximum\_speed(x) < 190, \\ \dfrac{maximum\_speed(x) - 190}{40} & \text{if } 190 \le maximum\_speed(x) \le 230, \\ 1 & \text{if } maximum\_speed(x) > 230, \end{cases}$$

where x is a variable corresponding to a car and $maximum\_speed(x)$ is a function which maps from the set of cars to *maximum_speed*.

The values of μ_{speedy_car} are shown in Table 2. They correspond to credibility of the statement that car x characterized by some *maximum_speed* is speedy.

From Table 2, we can induce the following functions $f^+(\cdot)$ and $g^+(\cdot)$,

$$f^+(\alpha) = \begin{cases} 0.2 & \text{if } 0 \le \alpha < 0.25, \\ 0.4 & \text{if } 0.25 \le \alpha < 0.5, \\ 0.6 & \text{if } 0.5 \le \alpha < 0.75, \\ 0.9 & \text{if } 0.75 \le \alpha \le 1, \end{cases} \quad \text{and} \quad g^+(\alpha) = \begin{cases} 0.3 & \text{if } \alpha = 0, \\ 0.8 & \text{if } 0 < \alpha \le 0.5, \\ 0.9 & \text{if } 0.5 < \alpha \le 0.75, \\ 1 & \text{if } 0.75 < \alpha \le 1. \end{cases}$$

Table 2. Degrees of membership to $X = \mathit{speedy\_car}$, $Y = \mathit{acceptable\_car}$.

Car	A	B	C	D	E	F	G	H	I	J
μ_{speedy_car}	0	0.25	0.5	0.25	0	1	0.5	0.75	0.25	1
$\mu_{acceptable\_car}$	0.3	0.4	0.6	0.8	0.2	0.9	0.7	0.9	0.7	1

Using functions $f^+(\cdot)$ and $g^+(\cdot)$, we can represent knowledge contained in Table 2 in terms of LP- and UP-decision rules having the following syntax:

- LP-rule: "if x is *speedy_car* with credibility $\mu_{speedy\_car}(maximum\_speed(x)) \geq \alpha$, then x is *acceptable_car* with credibility $f^+(\alpha) \geq \beta$";
- UP-rule: "if x is *speedy_car* with credibility $\mu_{speedy\_car}(maximum\_speed(x)) \leq \alpha$, then x is *acceptable_car* with credibility $g^+(\alpha) \leq \beta$".

The particular LP-rules induced from Table 2 are:

- "if $\mu_{speedy\_car}(maximum\_speed(x)) \geq 0$, then $\mu_{acceptable\_car}(x) \geq 0.2$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \geq 0.25$, then $\mu_{acceptable\_car}(x) \geq 0.4$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \geq 0.5$, then $\mu_{acceptable\_car}(x) \geq 0.6$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \geq 0.75$, then $\mu_{acceptable\_car}(x) \geq 0.9$".

The particular UP-rules induced from Table 2 are:

- "if $\mu_{speedy\_car}(maximum\_speed(x)) \leq 0$, then $\mu_{acceptable\_car}(x) \leq 0.3$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \leq 0.5$, then $\mu_{acceptable\_car}(x) \leq 0.8$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \leq 0.75$, then $\mu_{acceptable\_car}(x) \leq 0.9$",
- "if $\mu_{speedy\_car}(maximum\_speed(x)) \leq 1$, then $\mu_{acceptable\_car}(x) \leq 1$".

The syntax of LP- and UP-rules is similar to the syntax of "at least" and "at most" decision rules induced from dominance-based rough approximations of preference-ordered decision classes (Greco, Matarazzo, Słowiński [11, 14, 15]).

On the other hand, the syntax of LP- and UP-rules is more general than that of usual gradual rules introduced in (Dubois, Prade [6]). Indeed, while the usual gradual rules are statements of the type "if $\mu_X(x) \geq \alpha$, then $\mu_Y(x) \geq \alpha$", the LP-rules state "if $\mu_X(x) \geq \alpha$, then $\mu_Y(x) \geq \beta$". Therefore, the latter permits to consider different credibility of premise and conclusion.

The above decision rules can be modeled using Gaines-Rescher implication as follows:

- an LP-rule "if $\mu_X(x) \geq \alpha$, then $\mu_Y(x) \geq \beta$" is true with the credibility $I(f^+(\mu_X(x)), \mu_Y(x))$, where

$$I(a,b) = \begin{cases} 1 \text{ if } a \leq b \\ 0 \text{ otherwise} \end{cases} \quad \text{is the Gaines-Rescher implication;}$$

- an LN-rule "if $\mu_X(x) \leq \alpha$, then $\mu_Y(x) \geq \beta$" is true with the credibility $I(f^-(\mu_X(x)), \mu_Y(x))$;
- a UP-rule "if $\mu_X(x) \leq \alpha$, then $\mu_Y(x) \leq \beta$" is true with the credibility $I(\mu_Y(x), g^+(\mu_Y(x)))$;

- a UN-rule "if $\mu_X(x) \geq \alpha$, then $\mu_Y(x) \leq \beta$" is true with the credibility $I(\mu_Y(x), g^-(\mu_Y(x)))$.

For example, an LP-rule says that $\mu_Y(x) \geq f^+(\mu_X(x))$ for any pair of objects x and y. Remark that the implication used for modeling the above four gradual rules is based on ordinal properties of its arguments which are membership degrees of premise and conclusion. Moreover, the output of the implication is crisp, i.e. either 1 (true) or 0 (false).

The usual gradual rules can also be expressed in terms of the Gaines-Rescher implication, however, as these rules involve a single level of credibility α on the premise and on the conclusion, the rule "if $\mu_X(x) \geq \alpha$, then $\mu_Y(x) \geq \alpha$" is true with credibility $I(\mu_X(x), \mu_Y(x))$. When credibility of premise and conclusion is different, say α and β, then in the implication $I(\cdot,\cdot)$ we need $f^+(\mu_X(x))$ instead of $\mu_X(x)$ only.

Finally, let us observe that functions $f^+(\alpha)$, $f^-(\alpha)$, $g^+(\alpha)$ and $g^-(\alpha)$ can be seen as modifiers (Zadeh [28]) that give restrictions on credibility of the conclusion on the basis of credibility of the premise of a decision rule. Let us also mention that Bouchon-Meunier and Yao [4] studied fuzzy rules in relation with general modifiers and that Inuiguchi et al. [17] proposed to use modifiers for specification of possibility and necessity measures useful for induction of fuzzy rules.

3 Fuzzy Rough Approximations as a Basis for Rule Induction

Functions $f^+(\cdot)$, $f^-(\cdot)$, $g^+(\cdot)$ and $g^-(\cdot)$ are related to specific definitions of lower and upper approximations considered within rough set theory (Pawlak [21]). Let us consider a universe of the discourse U and two fuzzy sets, X and Y, defined on U by means of membership functions $\mu_X : U \to [0,1]$ and $\mu_Y : U \to [0,1]$. Suppose that we want to approximate knowledge contained in Y using knowledge about X. Let us also adopt the hypothesis that X is positively related to Y, which means "the more x is in X, the more x is in Y".

Then, the lower approximation of Y, given the information on X, is a fuzzy set $\underline{App}^+(X,Y)$, whose membership function for each $x \in U$, denoted by $\mu[\underline{App}^+(X,Y), x]$, is defined as follows,

$$\mu[\underline{App}^+(X,Y), x] = \inf_{z \in U : \mu_X(z) \geq \mu_X(x)} \{\mu_Y(z)\}. \tag{1}$$

Interpretation of the lower approximation (1) is based on a specific meaning of the concept of ambiguity. According to knowledge about X, the membership of object $w \in U$ to fuzzy set Y is ambiguous at the level of credibility ρ if there exists an object $z \in U$ such that $\mu_X(w) < \mu_X(z)$, however, $\mu_Y(w) \geq \rho$ while $\mu_Y(z) < \rho$. Thus, on the basis of knowledge about X and taking into account positive relationship between X and Y, x belongs to fuzzy set Y without ambiguity for any credibility level $\rho \leq \mu[\underline{App}^+(X,Y), x]$.

Remark that the above meaning of ambiguity is concordant with the dominance principle introduced to rough set theory in order to deal with preference-ordered data (Greco, Matarazzo, Słowiński [11, 14, 15]). In this case, the dominance principle says that, having an object with some membership degrees to X and Y, its modification consisting in an increase of its membership to X should not decrease its membership to Y; otherwise, the original object and the modified object are ambiguous.

Analogously, the upper approximation of Y given the information on X is a fuzzy set $\overline{App}^{+}(X,Y)$, whose membership function for each $x \in X$, denoted by $\mu[\overline{App}^{+}(X,Y),x]$, is defined as follows,

$$\mu[\overline{App}^{+}(X,Y),x] = \sup_{z \in U : \mu_X(z) \leq \mu_X(x)} \{\mu_Y(z)\}. \tag{2}$$

Interpretation of upper approximation (2) is based again on the above meaning of ambiguity. On the basis of knowledge about X and taking into account the positive relationship between X and Y, x belongs to fuzzy set Y with some possible ambiguity at credibility level $\rho \leq \omega$, where ω is the greatest membership degree of $z \in U$ to Y, such that ω is exactly equal to $\mu[\overline{App}^{+}(X,Y),x]$. In other words, even if $\mu_Y(x)$ would be smaller than ω, then due to the fact that there exists $z \in U$ with $\mu_X(z) \leq \mu_X(x)$, but $\mu_Y(z) = \omega > \mu_Y(x)$ (i.e., z and x are ambiguous), another object w, such that $\mu_X(w) = \mu_X(x)$, could belong to Y with credibility ω.

Example A (cont.). According to Defs. (1) and (2), the lower and the upper approximations of the set of cars described in Table 1 are presented in Table 3.

Table 3. Rough approximations of the set of cars
(X = *speedy_car*, Y = *acceptable_car*)

Car	A	B	C	D	E	F	G	H	I	J
μ_{speedy_car}	0	0.25	0.5	0.25	0	1	0.5	0.75	0.25	1
$\mu_{acceptable\_car}$	0.3	0.4	0.6	0.8	0.2	0.9	0.7	0.9	0.7	1
$\mu[\underline{App}^{+}(X,Y),x]$	0.2	0.4	0.6	0.4	0.2	0.9	0.6	0.9	0.4	0.9
$\mu[\overline{App}^{+}(X,Y),x]$	0.8	0.8	0.8	0.8	0.3	1	0.8	0.9	0.8	1

The concept of ambiguity and its relation with rough approximations can be illustrated on cars from Table 3. Let us consider car G. Its acceptability is 0.7. However, there is another car, C, having not smaller membership to *speedy_car* than G, but a smaller membership degree to *acceptable_car* than G. Therefore, acceptability 0.7 for car G is ambiguous. The highest membership degree to *acceptable_car* for which there is no ambiguity is equal to 0.6. Therefore, this value is the degree of membership of car G to the lower approximation of the set *acceptable_car*. Considering again car G, let us remark that there exists another car, D, having not greater membership degree to *speedy_car* than

G, but a greater membership degree to *acceptable_car* than G. Therefore, G is *acceptable_car* with the membership 0.8 which is possible for the reason of ambiguity between G and D. This is also the highest degree of membership of car G to *acceptable_car* taking into account the ambiguity; in consequence, it is the membership degree of car G to the upper approximation of *acceptable_car*.

These remarks on the concept of rough approximation can be summarized by saying that for each car in Table 3, degrees of membership to the lower and to the upper approximation of *acceptable_car*, respectively, give for each car the maximum degree of membership to *acceptable_car* without ambiguity and the maximum degree of membership to *acceptable_car* admitting some possible ambiguity.

The above definitions of fuzzy rough approximations are similar to definitions of fuzzy lower and upper approximations in terms of α-cuts proposed first by Nakamura and Gao [20]. They considered a fuzzy similarity relation on U and α-cuts on this relation, and defined a membership degree to lower or upper approximation for each level α and for each object $x \in U$, using the *inf* and *sup* operators, respectively, on objects $y \in U$ being similar to x in at least degree α. Our use of *sup* and *inf* in the definitions (1) and (2) is very similar to theirs, however, we do not consider any fuzzy similarity relation and the obtained membership degrees to lower or upper approximation depend only on object $x \in U$ and not on the specific level of the α-cut. In this sense, our definitions are non-parametric.

Let us comment on Defs.(1) and (2). Remark that, according to (1), we may have situation where $\mu_X(x) = 0$ and, nevertheless, $\mu[\underline{App}^+(X,Y),x] > 0$; in the above example, this corresponds to situation of car E. Analogously, according to (2), we may have situation where $\mu_X(x) = 1$ and, nevertheless, $\mu[\overline{App}^+(X,Y),x] < 1$. These situations are justified if one considers the knowledge about X as a piece of knowledge about Y; thus even if an object x has a zero membership in X, it might have a non-zero membership in Y; analogously, even if object x belongs to X with degree of membership equal to one, it might have a lower membership in Y. The above interpretation is coherent with the fact that, in our case, membership functions $\mu[\underline{App}^+(X,Y),x]$ and $\mu[\overline{App}^+(X,Y),x]$ are calculated from data. In other words, we adopt the hypothesis of "closed world" for these calculations. If we discard this interpretation, then for $\mu_X(x) = 0$, $\mu[\underline{App}^+(X,Y),x] = 0$, and for $\mu_X(x) = 1$, $\mu[\overline{App}^+(X,Y),x] = 1$, i.e., Defs. (1) and (2) take the form:

$$\mu[\underline{App}^+(X,Y),x] = \begin{cases} \inf\limits_{z \in U:\mu_X(z) \geq \mu_X(x)} \{\mu_Y(z)\} & \text{if } \mu_X(x) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{1'}$$

$$\mu[\overline{App}^+(X,Y),x] = \begin{cases} \sup\limits_{z \in U:\mu_X(z) \leq \mu_X(x)} \{\mu_Y(z)\} & \text{if } \mu_X(x) < 1, \\ 1 & \text{otherwise,} \end{cases} \tag{2'}$$

Remark, however, that for infinitesimal increase of $\mu_X(x) = 0$, or for infinitesimal decrease of $\mu_X(x) = 1$, Defs. (1') and (2') boil down to (1) and (2), respectively.

Table 4. 2^{nd} decision maker's evaluation of sample cars

Car	A	B	C	D	E	F	G	H	I	J
Mileage (km/l)	12	12	13	14	15	9	11	8	14	13
Acceptability	0.6	0.5	0.6	0.8	0.9	0.3	0.5	0.3	0.8	0.6

Table 5. Membership degrees of lower and upper approximations $\underline{App}^{+}(X,Y)$ and $\overline{App}^{+}(X,Y)$

Car	A	B	C	D	E	F	G	H	I	J
$\mu[\underline{App}^{+}(X,Y),x]$	0.5	0.5	0.6	0.8	0.9	0.3	0.5	0.3	0.8	0.6
$\mu[\overline{App}^{+}(X,Y),x]$	0.6	0.6	0.6	0.8	0.9	0.3	0.5	0.3	0.8	0.6

Example B. Let us consider another data set concerning the car selection problem, shown in Table 4.

We may define a fuzzy set X of *gas_saving_car* by the following membership function,

$$\mu_{gas\_saving\_car}(x) = \begin{cases} 0 & \text{if } mileage(x) < 9, \\ \dfrac{mileage(x) - 9}{6} & \text{if } 9 \le mileage(x) \le 15, \\ 1 & \text{if } mileage(x) > 15. \end{cases}$$

It has the following interpretation: "a car is definitely *gas_saving_car* if its mileage is greater than 15", and "a car is definitely not *gas_saving_car* if its mileage is smaller than 9". We may define a fuzzy set Y of *acceptable_car* using the degrees of acceptance from Table 4 as its membership grades.

Then, the lower and upper approximations $\underline{App}^{+}(X,Y)$ and $\overline{App}^{+}(X,Y)$ have the membership degrees shown in Table 5.

On the contrary, we may define a fuzzy set Z of *gas_guzzler_car* by a membership function $\mu_{gas\_guzzler\_car}(x) = 1 - \mu_{gas\_saving\_car}(x)$. Clearly, set Z is a complement of set X. In this case, the lower and upper approximations $\underline{App}^{+}(Z,Y)$ and $\overline{App}^{+}(Z,Y)$ have the following constant membership functions:

$$\mu[\underline{App}^{+}(Z,Y),x] = 0.3,$$
$$\mu[\overline{App}^{+}(Z,Y),x] = 0.9.$$

Those approximations are not very useful, however. □

From Example B, we recognize that the definitions of $\underline{App}^{+}(X,Y)$ and $\overline{App}^{+}(X,Y)$ are useful under the assumption that X is positively related with Y, i.e., when we suppose that "the more $\mu_Y(x)$, the more $\mu_X(x)$" or "the less $\mu_Y(x)$, the less $\mu_X(x)$".

When X is negatively related with Y, then "the more $\mu_Y(x)$, the less $\mu_X(x)$" or "the less $\mu_Y(x)$, the more $\mu_X(x)$". In example B, this corresponds to relationship between *gas_guzzler_car* and *acceptable_car*. We were able to see that in this case the definitions of $\underline{App}^{+}(X,Y)$ and $\overline{App}^{+}(X,Y)$ are not useful, thus we need another type of approximations.

Table 6. Membership degrees of lower and upper approximations $\underline{App}^-(Z,Y)$ and $\overline{App}^-(Z,Y)$

Car	A	B	C	D	E	F	G	H	I	J
$\mu[\underline{App}^-(Z,Y),x]$	0.5	0.5	0.6	0.8	0.9	0.3	0.5	0.3	0.8	0.6
$\mu[\overline{App}^-(Z,Y),x]$	0.6	0.6	0.6	0.8	0.9	0.3	0.5	0.3	0.8	0.6

The following definitions of the lower and upper approximations are appropriate when X is negatively related with Y,

$$\mu[\underline{App}^-(X,Y),x] = \inf_{z\in U:\mu_X(z)\leq\mu_X(x)} \{\mu_Y(z)\}, \tag{3}$$

$$\mu[\overline{App}^-(X,Y),x] = \sup_{z\in U:\mu_X(z)\geq\mu_X(x)} \{\mu_Y(z)\}. \tag{4}$$

Defs. (3) and (4) have similar interpretation in terms of ambiguity as Defs. (1) and (2), and they are also concordant with the dominance principle. Moreover, analogous remarks leading to Defs. ($1'$) and ($2'$) can be made for Defs. (3) and (4).

Of course, approximations (3) and (4) are not useful when X is positively related with Y. When X is not dependent on Y, it is not worthwhile approximating Y by X.

Example B (cont.). For $X = gas\_saving\_car$ and $Y = acceptable\_car$, the lower and upper approximations $\underline{App}^-(X,Y)$ and $\overline{App}^-(X,Y)$ have the following constant membership functions,

$$\mu[\underline{App}^-(X,Y),x] = 0.3,$$
$$\mu[\overline{App}^-(X,Y),x] = 0.9.$$

Those approximations are not very useful. On the other hand, if $Z = gas\_guzzler\_car$, then the lower and upper approximations $\underline{App}^-(Z,Y)$ and $\overline{App}^-(Z,Y)$ have the membership degrees shown in Table 6.

Comparing Table 6 with Table 5, we find that $\underline{App}^-(Z,Y) = \underline{App}^+(X,Y)$ and $\overline{App}^-(Z,Y) = \overline{App}^+(X,Y)$. This equality will be proved generally in the second part of the following theorem.

Theorem 1. The following properties are satisfied,

1) for each fuzzy set X and Y defined on U, and for each $x \in U$,

   1.1) $\mu[\underline{App}^+(X,Y),x] \leq \mu_Y(x) \leq \mu[\overline{App}^+(X,Y),x]$,

   1.2) $\mu[\underline{App}^-(X,Y),x] \leq \mu_Y(x) \leq \mu[\overline{App}^-(X,Y),x]$,

2) for any negation $N(\cdot)$, being a function $N : [0,1] \to [0,1]$ decreasing and such that $N(1) = 0$, $N(0) = 1$, for each fuzzy set X and Y defined on U, and for each $x \in U$,

$$\begin{aligned}
\text{2.1)}\quad \mu[\underline{App}^{+}(X,Y^{\mathrm{c}}),x] &= N(\mu[\overline{App}^{-}(X,Y),x]) \\
&= N(\mu[\overline{App}^{+}(X^{\mathrm{c}},Y),x]) = \mu[\underline{App}^{-}(X^{\mathrm{c}},Y^{\mathrm{c}}),x], \\
\text{2.2)}\quad \mu[\overline{App}^{+}(X,Y^{\mathrm{c}}),x] &= N(\mu[\underline{App}^{-}(X,Y),x]) \\
&= N(\mu[\underline{App}^{+}(X^{\mathrm{c}},Y),x]) = \mu[\overline{App}^{-}(X^{\mathrm{c}},Y^{\mathrm{c}}),x], \\
\text{2.1)}\quad \mu[\underline{App}^{-}(X,Y^{\mathrm{c}}),x] &= N(\mu[\overline{App}^{+}(X,Y),x]) \\
&= N(\mu[\overline{App}^{-}(X^{\mathrm{c}},Y),x]) = \mu[\underline{App}^{+}(X^{\mathrm{c}},Y^{\mathrm{c}}),x], \\
\text{2.2)}\quad \mu[\overline{App}^{-}(X,Y^{\mathrm{c}}),x] &= N(\mu[\underline{App}^{+}(X,Y),x]) \\
&= N(\mu[\underline{App}^{-}(X^{\mathrm{c}},Y),x]) = \mu[\overline{App}^{+}(X^{\mathrm{c}},Y^{\mathrm{c}}),x],
\end{aligned}$$

where for a given fuzzy set W, the fuzzy set W^{c} is its complement defined by $\mu_{W^{\mathrm{c}}}(x) = N(\mu_W(x))$.

Proof. Since, obviously, $\mu_X(x) \geq \mu_X(x)$, we have

$$\text{(i)}\qquad \mu[\underline{App}^{+}(X,Y),x] = \inf_{z\in U:\mu_X(z)\geq\mu_X(x)} \{\mu_Y(z)\} \leq \mu_Y(x).$$

Analogously, since $\mu_X(x) \leq \mu_X(x)$, we have

$$\text{(ii)}\qquad \mu_Y(x) \leq \sup_{z\in U:\mu_X(z)\leq\mu_X(x)} \{\mu_Y(z)\} = \mu[\overline{App}^{+}(X,Y),x].$$

From (i) and (ii) we obtain 1.1). Property 1.2) can be proved similarly. Thus, we proved 1).

According to the above definition of rough approximation, we have

$$\begin{aligned}
\text{(iii)}\qquad & \mu[\underline{App}^{+}(X,Y^{\mathrm{c}}),x] = \inf_{z\in U:\mu_X(z)\geq\mu_X(x)} \{N(\mu_Y(z))\} \\
&= N\left(\sup_{z\in U:\mu_X(z)\geq\mu_X(x)} \{\mu_Y(z)\}\right) = N\left(\sup_{z\in U:N(\mu_X(z))\leq N(\mu_X(x))} \{\mu_Y(z)\}\right) \\
&= \inf_{z\in U:N(\mu_X(z))\leq N(\mu_X(x))} \{N(\mu_Y(z))\}
\end{aligned}$$

and

$$\begin{aligned}
\text{(iv)}\qquad & N\left(\sup_{z\in U:\mu_X(z)\geq\mu_X(x)} \{\mu_Y(z)\}\right) = N(\mu[\overline{App}^{-}(X,Y),x]), \\
\text{(v)}\qquad & N\left(\sup_{z\in U:N(\mu_X(z))\leq N(\mu_X(x))} \{\mu_Y(z)\}\right) = N(\mu[\overline{App}^{+}(X^{\mathrm{c}},Y),x]), \\
\text{(iv)}\qquad & \inf_{z\in U:N(\mu_X(z))\leq N(\mu_X(x))} \{N(\mu_Y(z))\} = \mu[\underline{App}^{-}(X^{\mathrm{c}},Y^{\mathrm{c}}),x].
\end{aligned}$$

From (iii), (iv), (v) and (vi) we obtain 2.1). Properties 2.2)-2.4) can be proved analogously. □

Results 1) and 2) of Theorem 1 can be read as fuzzy counterparts of the following results well-known within the classical rough set theory: 1) says that fuzzy set Y includes its lower approximation and is included in its upper approximation; 2) represents a complementarity property – it says that the lower (upper) approximation of fuzzy set Y^c being positively related to X is the complement of the upper (lower) approximation of fuzzy set Y being negatively related with X (see first line of (2.1) and (2.2)); moreover, the lower (upper) approximation of fuzzy set Y^c being negatively related with X is the complement of the upper (lower) approximation of fuzzy set Y positively related with X (see first line of (2.3) and (2.4)). Result 2) of Theorem 1 states also other complementarity properties related to the complement of X (see second line of (2.1)-(2.4)); this complementarity property has not been considered within the classical rough set theory.

The lower and upper approximations defined above can serve to induce lower and upper credibility decision rules (called also certain and possible credibility decision rules) in the following way. Let us remark that inferring these decision rules is equivalent to finding functions $f^+(\cdot)$, $f^-(\cdot)$, $g^+(\cdot)$ and $g^-(\cdot)$.

These functions are defined as follows: for each $\alpha \in [0,1]$,

$$f^+(\alpha) = \begin{cases} \sup\limits_{\mu_X(x)\le\alpha} \{\mu[\underline{App}^+(X,Y),x]\} & \text{if } \exists x \in U : \mu_X(x) \le \alpha, \\ 0 & \text{otherwise,} \end{cases}$$

$$f^-(\alpha) = \begin{cases} \sup\limits_{\mu_X(x)\ge\alpha} \{\mu[\underline{App}^-(X,Y),x]\} & \text{if } \exists x \in U : \mu_X(x) \ge \alpha, \\ 0 & \text{otherwise,} \end{cases}$$

$$g^+(\alpha) = \begin{cases} \inf\limits_{\mu_X(x)\ge\alpha} \{\mu[\overline{App}^+(X,Y),x]\} & \text{if } \exists x \in U : \mu_X(x) \ge \alpha, \\ 1 & \text{otherwise,} \end{cases}$$

and

$$g^-(\alpha) = \begin{cases} \inf\limits_{\mu_X(x)\le\alpha} \{\mu[\overline{App}^-(X,Y),x]\} & \text{if } \exists x \in U : \mu_X(x) \le \alpha, \\ 1 & \text{otherwise.} \end{cases}$$

Example C. To give a more intuitive explanation of the fuzzy rough approximation proposed in this paper, let us consider the following example in which a continuous relationship between fuzzy set X and fuzzy set Y is assumed. The relationship between the membership degrees of fuzzy sets X and Y is described by function $h : [0,1] \to [0,1]$ such that $\mu_Y(x) = h[\mu_X(x)]$. Function $h(\cdot)$ is presented in Fig. 1.

Let us approximate fuzzy set Y using the knowledge about fuzzy set X. This approximation permits to define LP- and UP-rules. The result is presented in Fig. 2. Let us remark that in all its domain, function $h(\cdot)$ is contained between functions $f^+(\cdot)$ and $g^+(\cdot)$. This is concordant with point (1.1) of Theorem 1 stating that the membership degree of fuzzy set Y (corresponding to a value of function $h(\cdot)$) is always greater than or equal to the membership degree of its

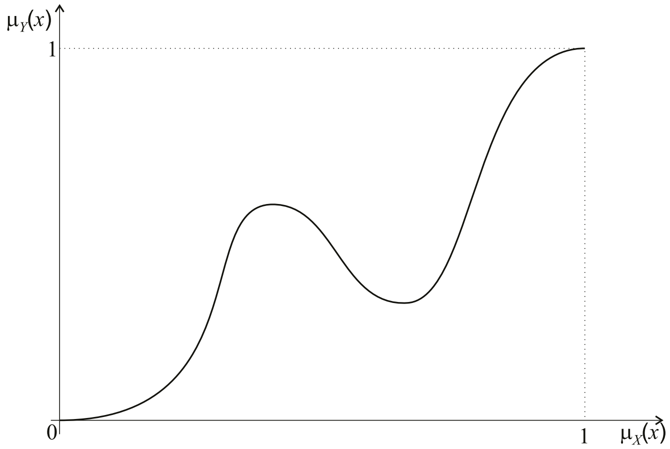


Fig. 1. Function $h(\cdot)$ representing the relationship between fuzzy sets X and Y.

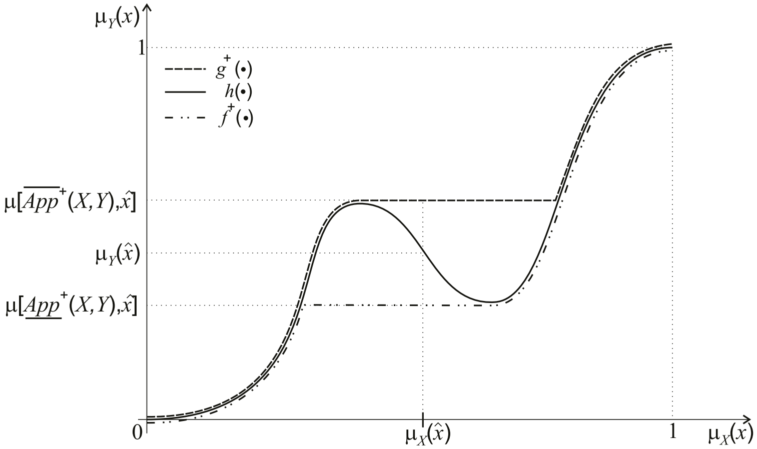


Fig. 2. Functions $f^+(\cdot)$ and $g^+(\cdot)$ representing, respectively, lower and upper approximations of fuzzy set Y subject to knowledge about fuzzy set X.

lower approximation (represented by a value of function $f^+(\cdot)$), and it is smaller than or equal to the membership degree of its upper approximation (represented by a value of function $g^+(\cdot)$).

Fig. 3 presents relationship between the membership degrees of two other fuzzy sets W and Z described by function $k : [0,1] \to [0,1]$ such that $\mu_Z(x) = k[\mu_W(x)]$.

Let us approximate fuzzy set Z using the knowledge about fuzzy set W. This approximation permits to define LN- and UN-rules. The result is presented in Fig. 4. Also in this case, function $k(\cdot)$ is contained between functions $f^-(\cdot)$ and $g^-(\cdot)$ in all its domain. This is concordant with point (1.2) of Theorem 1

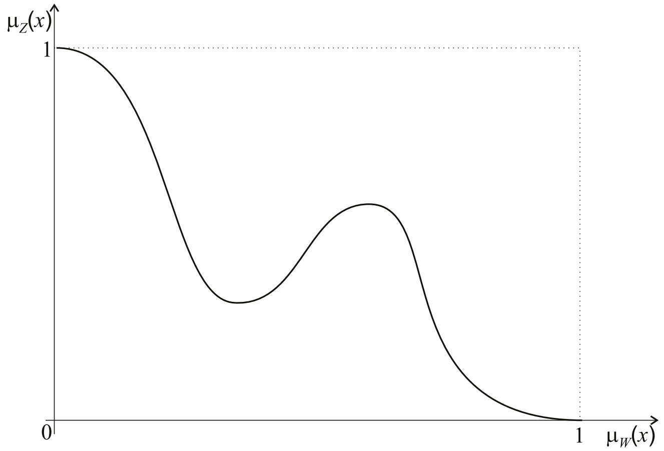


Fig. 3. Function $k(\cdot)$ representing the relationship between fuzzy sets W and Z.

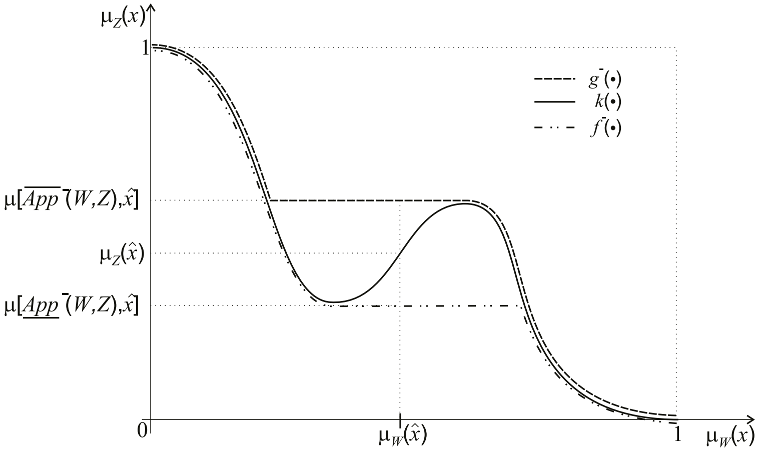


Fig. 4. Functions $f^-(\cdot)$ and $g^-(\cdot)$ representing, respectively, lower and upper approximations of fuzzy set Z subject to knowledge about fuzzy set W.

stating that the membership degree of fuzzy set Y (represented by a value of function $k(\cdot)$) is always greater than or equal to the membership degree of its lower approximation (represented by a value of function $f^-(\cdot)$), and it is smaller than or equal to the membership degree of its upper approximation (represented by a value of function $g^-(\cdot)$).

4 Fuzzy Rough Modus Ponens

While the previous sections concentrated on the issues of representation, rough approximation and decision rule induction, this section is devoted to inference with a generalized *modus ponens* (*MP*).

Classically, *MP* has the following form:

$$\begin{array}{lll} \text{if} & X \rightarrow Y & \text{is true} \\ \text{and} & X & \text{is true} \\ \hline \text{then} & Y & \text{is true} \end{array}$$

MP has the following interpretation: assuming a consequence relation $X \rightarrow Y$ (decision rule) and a fact X (premise), we obtain another fact Y (conclusion). If we replace the classical decision rule above by our four kinds of decision rules, then we obtain the following four generalized fuzzy-rough *MP*:

$$\left.\begin{array}{ll} \text{if} & \mu_X(x) \geq \alpha \rightarrow \mu_Y(x) \geq f^+(\alpha) \\ \text{and} & \mu_X(x) = \alpha' \\ \hline \text{then} & \mu_Y(x) \geq f^+(\alpha') \end{array}\right\} \qquad (\mathit{LP\text{-}MP})$$

$$\left.\begin{array}{ll} \text{if} & \mu_X(x) \leq \alpha \rightarrow \mu_Y(x) \geq f^-(\alpha) \\ \text{and} & \mu_X(x) = \alpha' \\ \hline \text{then} & \mu_Y(x) \geq f^-(\alpha') \end{array}\right\} \qquad (\mathit{LN\text{-}MP})$$

$$\left.\begin{array}{ll} \text{if} & \mu_X(x) \leq \alpha \rightarrow \mu_Y(x) \leq g^+(\alpha) \\ \text{and} & \mu_X(x) = \alpha' \\ \hline \text{then} & \mu_Y(x) \leq g^+(\alpha') \end{array}\right\} \qquad (\mathit{UP\text{-}MP})$$

$$\left.\begin{array}{ll} \text{if} & \mu_X(x) \geq \alpha \rightarrow \mu_Y(x) \leq g^-(\alpha) \\ \text{and} & \mu_X(x) = \alpha' \\ \hline \text{then} & \mu_Y(x) \leq g^-(\alpha') \end{array}\right\} \qquad (\mathit{UN\text{-}MP})$$

The above generalized fuzzy-rough *MP*s suppose that the premise is a crisp fact, thus the conclusion is also a crisp fact. Precisely, the crisp facts mean punctual membership degrees of x in X and x in Y. Remark that the generalized fuzzy-rough *MP*s allow the inference with a fuzzy fact as premise, leading to a fuzzy fact as conclusion. In this case, the fuzzy facts mean that the membership degrees of x in X and x in Y are fuzzy numbers. These *MP*s will be called generalized fuzzy^2-rough *MP*s.

To explain this issue in more detail, let $\mu_X(x) = \tilde{\alpha}$ be a fuzzy premise in all four generalized *MP*s. Let also start the explanation by fuzzy^2-rough *LP-MP*. If we know crisp $\mu_X(x) = \alpha$, we obtain crisp conclusion $\mu_Y(x) \geq f^+(\alpha) = \beta$ from the fuzzy-rough *LP-MP*. Having the information $\mu_X(x) = \tilde{\alpha}$, a membership grade $\mu_{\tilde{\alpha}}(\alpha)$ is assigned to the premise $\mu_X(x) = \alpha$. Let $\tilde{\beta}$ be a fuzzy number corresponding to the credibility of the conclusion $\mu_Y(x) \geq \beta$. For given $\mu_{\tilde{\alpha}}(\alpha)$, we are considering the conclusion $\mu_Y(x) \geq f^+(\alpha)$; its credibility $\mu_{\tilde{\beta}}(\beta)$, where $\beta = f^+(\alpha)$, obviously depends on $\mu_{\tilde{\alpha}}(\alpha)$. This dependency should observe the following principle of coherence: considering two possible conclusions, "$\mu_Y(x) \geq \beta_1$" and "$\mu_Y(x) \geq \beta_2$", if $\beta_1 \leq \beta_2$, then $\mu_{\tilde{\beta}}(\beta_1) \geq \mu_{\tilde{\beta}}(\beta_2)$; for example, if "$\mu_Y(x) \geq \beta_1$" means $\mu_{acceptable\_car}(x) \geq 0.5$ and "$\mu_Y(x) \geq \beta_2$" means $\mu_{acceptable\_car}(x) \geq 0.7$, then, according to *LP-MP*, we cannot assign to the first conclusion a credibility smaller than that of the second conclusion, because otherwise, it would mean that we give a smaller credit to the less demanding conclusion. Indeed, it would be quite paradoxical to conclude that we have credibility, say 1, for $\mu_{acceptable\_car}(x) \geq 0.7$ and only 0.6 for $\mu_{acceptable\_car}(x) \geq 0.5$: this would mean that it is sure for us that $\mu_{acceptable\_car}(x)$ is greater than or equal to 0.7, but we have some doubts if $\mu_{acceptable\_car}(x)$ is greater than of equal to 0.5.

Varying $\alpha \in [0,1]$, we get a family of conclusions $\mu_Y(x) \geq \beta = f^+(\alpha)$. According to the above principle of coherence, the credibility $\tilde{\beta}$ of each conclusion is a fuzzy number whose membership function $\mu_{\tilde{\beta}}(\cdot)$ is defined as follows,

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : f^+(\alpha) \geq \beta\} & \text{if } \exists\alpha \text{ such that } f^+(\alpha) \geq \beta, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(i)}$$

In all other cases, a similar reasoning gives the following membership function $\mu_{\tilde{\beta}}(\cdot)$,

- for fuzzy^2-rough *LN-MP*:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : f^-(\alpha) \geq \beta\} & \text{if } \exists\alpha \text{ such that } f^-(\alpha) \geq \beta, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(ii)}$$

- for fuzzy^2-rough *UP-MP*:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : g^+(\alpha) \leq \beta\} & \text{if } \exists\alpha \text{ such that } g^+(\alpha) \leq \beta, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(iii)}$$

- for fuzzy^2-rough *UN-MP*:

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} \sup\{\mu_{\tilde{\alpha}}(\alpha) : g^-(\alpha) \leq \beta\} & \text{if } \exists\alpha \text{ such that } g^-(\alpha) \leq \beta, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(iv)}$$

It is quite natural to assume that the credibility of the premise in a fuzzy^2-rough *MP* is represented by an *LR*-fuzzy number, i.e. convex, normal, continuous fuzzy subset of the real interval $[0,1]$. An *LR*-fuzzy number $\tilde{\alpha}$ is represented by the 6-tuple $(\alpha^L, \alpha^U, \Delta_L, \Delta_U, L, R)$, with $\alpha^L \leq \alpha^U$, $\Delta_L \geq 0$, $\Delta_U \geq 0$, and

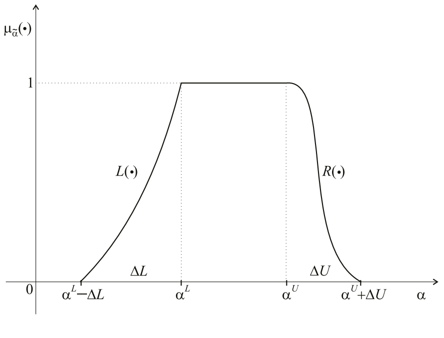


Fig. 5. *LR*-fuzzy number.

L, R being non-increasing functions in $[0,1]$ for which $L(0) = R(0) = 1$ and $L(1) = R(1) = 0$, such that the membership function of $\tilde{\alpha}$ is:

$$\mu_{\tilde{\alpha}}(\alpha) = \begin{cases} L((\alpha^L - \alpha)/\Delta_L) & \text{if } \alpha < \alpha^L \text{ and } \Delta_L > 0, \\ 1 & \text{if } \alpha^L \leq \alpha \leq \alpha^U, \\ R((\alpha - \alpha^U)/\Delta_U) & \text{if } \alpha > \alpha^U \text{ and } \Delta_U > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The membership function $\mu_{\tilde{\alpha}}(\cdot)$ can be interpreted as follows: $[\alpha^L, \alpha^U]$ is an interval of the "most possible" values and Δ_L and Δ_U are non-negative left and right "spreads" of $\tilde{\alpha}$ (see Fig. 5).

Taking into account (i) to (iv), when the premise is an *LR*-fuzzy number, the conclusion in each of generalized fuzzy^2-rough *MP*s is as follows:

- for *LP-MP*,

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} 1 & \text{if } f^+(\alpha^U) \geq \beta, \\ R((f^{+*}(\beta) - \alpha^U)/\Delta_U) & \text{if } f^+(\alpha^U) < \beta \leq f^+(\alpha^U + \Delta_U) \text{ and } \Delta_U > 0, \\ 0 & \text{otherwise,} \end{cases}$$

  where, if $\Delta_U > 0$, then, for each $\beta \in]f^+(\alpha^U), f^+(\alpha^U + \Delta_U)]$,

$$f^{+*}(\beta) = \inf\{\alpha \in [0,1] : f^+(\alpha) \geq \beta\},$$

- for *LN-MP*,

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} 1 & \text{if } f^-(\alpha^L) \geq \beta, \\ L((\alpha^L - f^{-*}(\beta))/\Delta_L) & \text{if } f^-(\alpha^L) < \beta \leq f^-(\alpha^L - \Delta_L) \text{ and } \Delta_L > 0, \\ 0 & \text{otherwise,} \end{cases}$$

  where, if $\Delta_L > 0$, then, for each $\beta \in]f^-(\alpha^L), f^-(\alpha^L - \Delta_L)]$,

$$f^{-*}(\beta) = \sup\{\alpha \in [0,1] : f^-(\alpha) \geq \beta\},$$

- for *UP-MP*,

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} 1 & \text{if } g^+(\alpha^L) \leq \beta, \\ L((\alpha^L - g^{+*}(\beta))/\Delta_L) & \text{if } g^+(\alpha^L - \Delta_L) \leq \beta < g^+(\alpha^L) \text{ and } \Delta_L > 0, \\ 0 & \text{otherwise,} \end{cases}$$

  where, if $\Delta_L > 0$, then, for each $\beta \in [g^+(\alpha^L - \Delta_L), g^+(\alpha^L)[$,

$$g^{+*}(\beta) = \sup\{\alpha \in [0,1] : g^+(\alpha) \leq \beta\},$$

- for *UN-MP*,

$$\mu_{\tilde{\beta}}(\beta) = \begin{cases} 1 & \text{if } g^-(\alpha^U) \leq \beta, \\ R((g^{-*}(\beta) - \alpha^U)/\Delta_U) & \text{if } g^-(\alpha^U + \Delta_U) \leq \beta < g^-(\alpha^U) \text{ and } \Delta_U > 0, \\ 0 & \text{otherwise,} \end{cases}$$

  where, if $\Delta_U > 0$, then, for each $\beta \in [g^-(\alpha^U + \Delta_U), g^-(\alpha^U)[$,

$$g^{-*}(\beta) = \inf\{\alpha \in [0,1] : g^-(\alpha) \leq \beta\}.$$

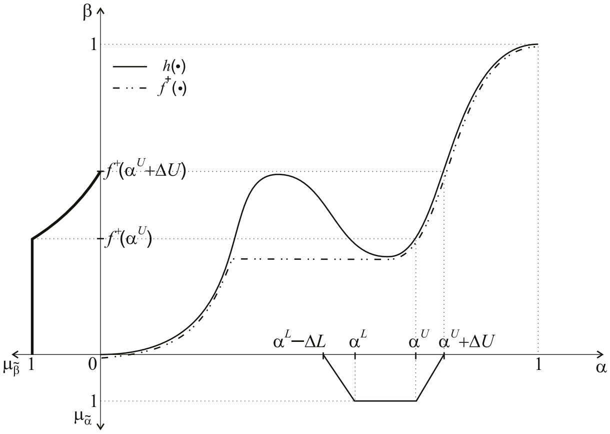


Fig. 6. Fuzzy^2-rough *LP-MP*.

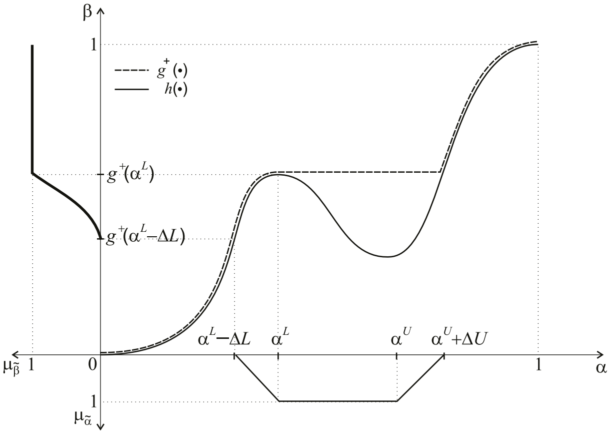


Fig. 7. Fuzzy^2-rough *UP-MP*.

Figs. 6 and 7 illustrate the generalized fuzzy^2-rough *LP-MP* and *UP-MP*. Generalized fuzzy^2-rough *LN-MP* and *UN-MP* have analogous interpretation.

Let us remark that applying the above four types of generalized fuzzy^2-rough *MP*s for each value β we get a membership degree for the hypothesis that the credibility of the conclusion (that is $\mu_Y(x)$) is greater than or equal to β, denoted by $\mu^+_{\tilde{\beta}}(\beta)$ (case of *LP-MP* or *LN-MP*), and a membership degree for the hypothesis that the credibility of the conclusion (that is $\mu_Y(x)$) is smaller than or equal to β, denoted by $\mu^-_{\tilde{\beta}}(\beta)$ (case of *UP-MP* or *UN-MP*). On the basis of the membership degrees $\mu^+_{\tilde{\beta}}(\beta)$ and $\mu^-_{\tilde{\beta}}(\beta)$, for each value of β, we can have the following four situations:

a) there is some possibility that credibility of the conclusion is greater than β and there is no possibility that the credibility of the conclusion is smaller than β, which means: $\mu^{+}_{\tilde{\beta}}(\beta) > 0$ and $\mu^{-}_{\tilde{\beta}}(\beta) = 0$,

b) there is no possibility that credibility of the conclusion is greater than β and there is some possibility that the credibility of the conclusion is smaller than β, which means: $\mu^{+}_{\tilde{\beta}}(\beta) = 0$ and $\mu^{-}_{\tilde{\beta}}(\beta) > 0$,

c) there is no possibility that credibility of the conclusion is greater than β and there is no possibility that the credibility of the conclusion is smaller than β, which means: $\mu^{+}_{\tilde{\beta}}(\beta) = 0$ and $\mu^{-}_{\tilde{\beta}}(\beta) = 0$,

d) there is some possibility that credibility of the conclusion is greater than β and there is some possibility that the credibility of the conclusion is smaller than β, which means: $\mu^{+}_{\tilde{\beta}}(\beta) > 0$ and $\mu^{-}_{\tilde{\beta}}(\beta) > 0$.

While situations a) and b) have a clear result, situations c) and d) have not. In situation c) we have not enough information neither to say that the credibility of the conclusion is greater than or equal to β nor to say that the credibility of the conclusion is smaller than or equal to β. In situation d) we have contradictory information that the credibility of the conclusion is simultaneously greater than or equal to β and smaller than or equal to β. These four situations correspond to four truth qualifications considered within the four-valued logic (Belnap [2]) and its fuzzy extensions proposed by Perny and Tsoukiàs [22], and by Fortemps and Słowiński [9] (for other applications of the four-valued logic within rough set theory see Greco et al. [16]). Indeed, $\mu^{+}_{\tilde{\beta}}(\beta)$ corresponds to an argument in favor of the proposition "*credibility of conclusion is at least* β", while $\mu^{-}_{\tilde{\beta}}(\beta)$ corresponds to an argument against the same proposition. Therefore,

- in situation a) the proposition is *true* (argument in favor and no argument against),
- in situation b) the proposition is *false* (no argument in favor and argument against),
- in situation c) the proposition is *unknown* (no argument in favor and no argument against – see Fig. 8),
- in situation d) the proposition is *contradictory* (argument in favor and argument against – see Fig. 9).

5 Conclusions and Further Research Directions

In this paper we presented a new fuzzy rough set approach. The main advantage of this new approach is that it infers the most cautious conclusions from available imprecise information, without using neither fuzzy connectives nor specific parameters, whose choice are always subjective to a certain extent. Another advantage of our approach is that it uses only the ordinal properties of membership degrees. We noticed that our approach is related to:

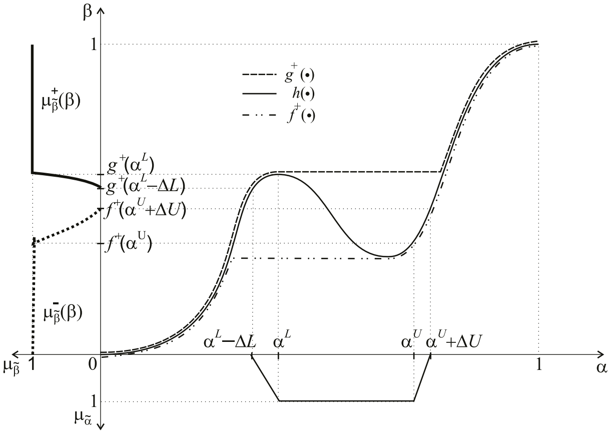


Fig. 8. Fuzzy^2-rough *LP-MP* and *UP-MP* with unknown proposition.

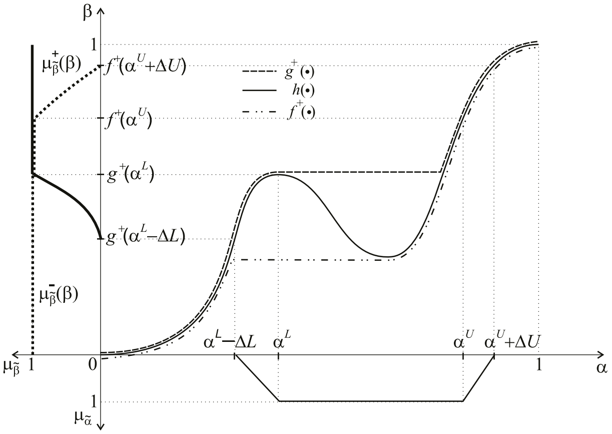


Fig. 9. Fuzzy^2-rough *LP-MP* and *UP-MP* with contradictory proposition.

– gradual rules, with respect to syntax and semantics of considered decision rules,
– fuzzy rough set approach based on α-cuts, with respect to the specific use of *sup* and *inf* in the definition of lower and upper approximations,
– dominance-based rough set approach, with respect to the idea of monotonic relationship between credibility of premise and conclusion.

We think that this approach gives a new prospect for applications of fuzzy rough approximations in real-world decision problems. More precisely, we envisage the following four extensions of this methodology:

1) Multi-premise generalization: in this paper we dealt with relationships between membership degrees of fuzzy sets X and Y, however, in real-world problems we are interested in relationships between membership degrees to fuzzy sets X_i, $i = 1, \ldots, n$, and Y_j, $j = 1, \ldots, m$, which leads to consideration of gradual decision rules of the type: "*the larger the market share of a company and the smaller its debt, the larger its profit and the better its financial rating*";
2) Variable-precision rough approximation: in this paper we propose to calculate the degree of membership to the fuzzy lower approximation on the basis of non-ambiguous objects only, however, it might be useful in practical applications to allow a limited number of ambiguous objects as well; in this way we may get less specific rules of the type: "*the larger the market share of a company, the larger its profit, in $l\%$ of the cases*", where l is a parameter controlling the proportion of ambiguous objects in the definition of the lower approximation.
3) Further generalization of *modus ponens*: in this paper, we introduced generalized fuzzy-rough *modus ponens*, so that we can infer a conclusion from a gradual rule and a crisp or fuzzy fact. We may further extend the fuzzy-rough *modus ponens* to infer a conclusion from multiple gradual rules, so that we obtain a method for approximate reasoning.
4) Imprecise input data: the evaluation table from which the rough approximations and gradual decision rules are induced may include imprecise or missing values; such values can be represented by fuzzy numbers.

Acknowledgements

The research of the first author has been supported by the Italian Ministry of Education, University and Scientific Research (MIUR). The third author wishes to acknowledge financial support from the State Committee for Scientific Research and from the Foundation for Polish Science, subsidy no. 11/2001.

References

1. Aragones, E., Gilboa, I., Postlewaite, A., Schmeidler, D., *From cases to rules: induction and regression*, 2002, manuscript
2. Belnap, N., A useful four-valued logic, in: *Modern Uses of Multiple Valued Logics*, D. Reidel, Dordrecht, 1977, pp.8–37
3. Bilgic, T., Turksen, I. B., Measurement-theoretic justification of connectives in fuzzy set theory, *Fuzzy Sets and Systems*, **76**(3) (1995), 289–308
4. Bouchon-Mounier, B., Yao, J., Linguistic modifiers and gradual membership to a category, *International Journal on Intelligent Systems*, **7** (1992) 26–36
5. Cattaneo, G., Fuzzy extension of rough sets theory, in: L. Polkowski, A. Skowron (eds.), *Rough Sets and Current Trends in Computing*, LNAI 1424, Springer-Verlag, Berlin 1998, pp.275–282
6. Dubois, D., Prade, H., Gradual inference rules in approximate reasoning, *Information Sciences*, **61** (1992) 103–122

7. Dubois, D. and Prade, H., Putting rough sets and fuzzy sets together, in: R. Słowiński (ed.), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, 1992, pp.203–232
8. Dubois, D., Prade, H., Yager, R., A Manifesto: Fuzzy Information Engineering, in: Dubois, D., Prade, H., Yager, R. (eds.), *Fuzzy Information Engineering*, J.Wiley, New York, 1997, pp.1–8
9. Fortemps, Ph., Słowiński, R., A graded quadrivalent logic for ordinal preference modelling: Loyola-like approach, *Fuzzy Optimization and Decision Making*, **1**(1) (2002) 93–111
10. Greco, S., Inuiguchi, M., Słowiński, R., Dominance-based rough set approach using possibility and necessity measures, in: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (eds.), *Rough Sets and Current Trends in Computing*, LNAI 2475, Springer-Verlag, Berlin, 2002, pp.85–92
11. Greco, S., Matarazzo, B., Słowiński R., The use of rough sets and fuzzy sets in MCDM. Chapter 14 in: T. Gal, T. Stewart, T. Hanne (eds.), *Advances in Multiple Criteria Decision Making*, Kluwer Academic Publishers, Boston, 1999, pp.14.1–14.59
12. Greco, S., Matarazzo, B., Słowiński, R., Rough set processing of vague information using fuzzy similarity relations, in: C.S. Calude and G. Paun (eds.), *Finite Versus Infinite – Contributions to an Eternal Dilemma*, Springer-Verlag, London, 2000a, pp.149–173
13. Greco, S., Matarazzo, B., Słowiński, R., Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting, in: J. Fodor, B. De Baets and P. Perny (eds.), *Preferences and Decisions under Incomplete Knowledge*, Physica-Verlag, Heidelberg, 2000b, pp.131–151
14. Greco, S., Matarazzo, B., Słowiński, R., Rough sets theory for multicriteria decision analysis. *European J. of Operational Research* **129**(1) (2001) 1–47
15. Greco, S., Matarazzo, B., Słowiński R., Multicriteria classification, in: W.Kloesgen and J.Żytkow (eds.), *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, New York, 2002, chapter 16.1.9, pp.318–328
16. Greco, S., Matarazzo, B., Słowiński, R., Tsoukias, A., Exploitation of a rough approximation of the outranking relation in multicriteria choice and ranking, in: T.J. Stewart and R.C. van den Honert (eds.), *Trends in Multicriteria Decision Making*. LNEMS 465, Springer-Verlag, Berlin, 1998, pp.45–60
17. Inuiguchi, M., Greco, S., Słowiński, R., Tanino, T., Possibility and necessity measure specification using modifiers for decision making under fuzziness, Fuzzy Sets and Systems **137**(1) (2003) 151–175
18. Inuiguchi, M., Tanino, T., New Fuzzy Rough Sets Based on Certainty Qualification, in S.K. Pal, L. Polkowski and A. Skowron (eds.): *Rough-Neural Computing: Techniques for Computing with Words*, Springer-Verlag, Berlin, 2002, pp.277–296
19. Nakamura, A., Applications of fuzzy-rough classification to logics, in: R. Słowiński (ed.), *Intelligent Decision Support: Handbook of Applications and Advances of the Sets Theory*, Kluwer, Dordrecht, 1992, pp.233–250
20. Nakamura, A., Gao, J.M., A logic for fuzzy data analysis, *Fuzzy Sets and Systems*, **39** (1991) 127–132
21. Pawlak, Z., *Rough Sets*, Kluwer, Dordrecht, 1991
22. Perny, P., Tsoukias, A., On the continuous extension of a four valued logic for preference modeling, in: *Proc. IPMU 1998*, pp.302–309
23. Polkowski, L., *Rough Sets: Mathematical Foundations*, Physica-Verlag, Heidelberg, 2002

24. Słowiński, R., Rough set processing of fuzzy information, in: T.Y. Lin, A. Wildberger (eds.), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery.* Simulation Councils, Inc., San Diego, CA, 1995, 142–145
25. Słowiński, R., Stefanowski, J., Rough set reasoning about uncertain data, *Fundamenta Informaticae* **27** (1996) 229–243
26. Wittgenstein, L., *Tractatus Logico-Philosophicus.* Routledge and Kegan Paul, London, 1922; fifth impression 1951
27. Yao, Y.Y., Combination of rough and fuzzy sets based on *alpha*-level sets, in: T.Y. Lin and N. Cercone (eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Kluwer, Boston, 1997, pp.301–321
28. Zadeh, L.A., A Fuzzy Set-Theoretic Interpretation of Linguistic Hedges, *J. Cybernetics*, **2** (1972) 4–34

Emergent Rough Set Data Analysis

Yasser Hassan[1] and Eiichiro Tazaki[1]

Department of Control and System Engineering, Toin University of Yokohama,
1614 Kurogane-cho, Aoba-ku, Yokohama 225-8502, Japan
{y_fouad,tazaki}@intlab.toin.ac.jp

Abstract. Many systems in nature produce complicated behaviors, which emerge from the local interactions of relatively simple individual components that live in some spatially extended world. Notably, this type of emergent behavior formation often occurs without the existence of a central control. The rough set concept is a new mathematical approach to imprecision, vagueness and uncertainty. This paper introduces the emergent computational paradigm and discusses its applicability and potential in rough sets theory. In emergence algorithm, the overall system dynamics emerge from the local interactions of independent objects or agents. For accepting a system is displaying an emergent behavior, the system should be constructed by describing local elementary interactions between components in different ways than those used in describing global behavior and properties of the running system over a period of time. The proposals of an emergent computation structure for implementing basic rough sets theory operators are also given in this paper.

Keywords: rough sets, emergent behavior, reduct, rule induction, discretization

1 Introduction

In recent years, computer models and knowledge-based systems have become important as aids for the management systems. In this methodology, the behavior of the overall software system arises (emerges) from the interactions of quasi-independent computational agents or objects [10]. Each agent contains the entire specification for its behavior, which includes interactions between it and its computational environment and other agents. The high-level behavior of the system is entirely the result of low-level interactions between agents and between agents and their local environments. Computational science is able to extend the benefits of experimental science into new phenomena that are not studied by traditional experimentation. The development of this computational approach can currently be seen in a wide range of academic disciplines [1, 4, 7–9, 14, 16].

Knowledge discovery in databases [6, 17] is a form of machine learning, which discovers interesting knowledge from database and represents the knowledge in a high-level language. Data mining is a main step in the multi-stage process of knowledge discovery [13]. It concentrates on exploration and analysis of large quantities of data in order to discover meaningful patterns and rules. Rough set methodology provides a powerful tool for knowledge discovery from large

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 343–361, 2004.

and incomplete dataset [12]. Rough set approach is based on generalization of the relation of being an element of a set. Calculation over basic indiscernibility relations attaches to each considered property lower and upper approximation, corresponding to elements that satisfy this property with certainty and which possibly satisfy it, respectively [13]. Then, a subset of objects with considered property is called rough with respect to a given knowledge about object indiscernibility, if its lower approximation is not equal to the upper one.

The purpose of rough set KDD [12] is to find models that describe or classify the measurement data. Such models may take on a wide variety of forms according to the model construction scheme used. The choice of modelling scheme bears both on the nature of the pattern recognition problem, as well as on the purpose of the modelling task. The goal here is to gain insight into the problem at hand by analyzing the constructed model, i.e., the structure of the model is itself of interest. The process of constructing a model is an iterated waterfall cycle with possible backtracking on the individual sub-steps. It is therefore important to have a set of tools available that render possible this type of flexible experimentation. However, a complete model construction and experimentation tool must comprise more than a collection of clever algorithms in order to be fully useful. It is needed to set the tools in an environment such that intermediate results can be viewed and analyzed, and decision for further processing made. Basically, an environment to interactively manage and process data is required.

This paper argues that rough set theory belongs to a unique class of problem solvers that display emergent behavior [1, 7–9], a method of problem solving allows task specific knowledge to emerge from the interactions of problem solver and a task description presented by information system. Demonstrations and guidelines showing how to exploit emergent intelligence [1] to extend the problem solving capabilities in rough sets and other soft computing are also presented. Hence, a problem addressed in this paper is to establish a certain background for constructing emergent system structure [7, 14], which might be able to perform basic rough set operations, in particular, those that reduce the redundancy of knowledge, if any. Such a system has to be equipped with facilities, which allows us to find fundamental rough set concepts and a strategy for the development of high performance solutions using emergent computation is illustrated in outline form for a problem domain.

The structure of the chapter is as follows. In Sect. 2, we present the notion of an emergent system and we construct rules to test if the system is emergence or not. Sect. 3 presents rough set preliminaries. Standard rough set algorithms for synthesis of decision rules from decision tables are described also. Sect. 4 presents the new technique of rough sets to enhance the emergent behavior in problem solving algorithm. Sect. 5 shows the guidelines to increase the observation of emergent phenomena in rough set methodology. Some applications of emergent rough set are presented in Sect. 6. The paper is concluded in Sect. 7.

2 Rough Set Theory

In this section we recall some basic notions related to information systems and rough sets. Rough set concept, introduced by Pawlak [11], one of its essential merits being its direct relation to classification problems, is founded on the assumption that each object is associated with some information (data, knowledge). Objects that are characterized by the same information are said to be indiscernible in the view of available data. This induces the indiscernibility relation (equivalence relation) that is the mathematical basis of rough set theory.

Knowledge representation in rough set data analysis is constructed via an information system. An information system [6] is a pair $IS = (U, C)$, where U is non-empty finite set of objects called the universe, and $C = \{c_1, c_2, \ldots, c_m\}$ is a non–empty finite set of mappings $c : U \rightarrow \nu_c$; each $c \in C$ is called a conditional attribute. Decision table $DT = (U, C \cup D)$ is a special case of information system introduced in rough set theory as tool to present data, where the major feature that differs it from the information system is the distinguished attribute set D, where $D \cap C = \emptyset$, called the decision attribute set.

Let $I \subseteq U \times U$ denote an equivalence relation of indiscernibility on U that is defined as,

$$I_B = \{(x, y) \in U \times U : b(x) = b(y),\ \forall b \in B\}, \tag{1}$$

where $B \subseteq C$. Objects x and y satisfying the relation I are indiscernible by attributes from B; the relation I is reflexive, symmetric, and transitive.

An ordered pair $AS = (U, I)$ is called an approximation space. According to I, we can define two crisp sets $\underline{B}X$ and $\overline{B}X$ called the lower and the upper approximation of the set of objects X in the approximation space AS,

$$\begin{array}{l} \underline{B}X = \{x \in U : I_B(x) \subseteq X\}, \quad \text{and} \\ \overline{B}X = \{x \in U : I_B(x) \cap X \neq \emptyset\}, \end{array} \tag{2}$$

where $I_B(x)$ denotes the set of all objects indiscernible with x, i.e., the equivalence class determined by x. Pawlak [11] defined a rough set to be a family of subsets of a universe that have the same lower and upper approximations.

Table 1 has an example of a decision table where the set of attributes $C = \{Sex,\ Clinical\ Stage\}$, the set of values of decision attribute is $V_{infection} = \{Yes,\ No\}$, and $U = \{P1, P2, \cdots, P10\}$.

Table 1. An example.

U	Sex	Clinical Stage	Infection
$P1$	F	T	Yes
$P2$	M	T	Yes
$P3$	F	B	No
$P4$	M	T	Yes
$P5$	M	T	Yes
$P6$	M	T	No
$P7$	F	T	Yes
$P8$	F	B	No
$P9$	M	B	No
$P10$	M	T	No

The upper and lower approximations of the decision attribute "*infection*" can be found as follows:

$$\begin{aligned}\underline{X}_{infection=yes} &= \{P1, P7\}\\ \overline{X}_{infection=yes} &= \{P1, P2, P4, P5, P6, P7, P10\}\\ \underline{X}_{infection=no} &= \{P3, P8, P9\}\\ \overline{X}_{infection=no} &= \{P2, P3, P4, P5, P6, P8, P9, P10\}\end{aligned}$$

One of the problems related to practical applications of rough set methods is whether the whole set of attributes is necessary and if not, how to determine the simplified and still sufficient subset of attributes equivalent to the original. The rejected attributes are redundant since their removal cannot worsen the classification. There are usually several such subsets of attributes and those, which are minimal with respect to inclusion, are called reducts.

Decision rules [12, 13] can be perceived as data patterns, which represent relationships between attribute values of a classification system. If $V = \bigcup\{\nu_c : c \in C\} \cup \nu_D$, is a set of values for attributes, then the decision rule is a logical formula: IF $(c_1 = \nu_1)$ $\& \cdots \&$ $(c_n = \nu_n)$ THEN $(d = \nu_d)$. The decision rule is true in the decision table if $\{u \in U : c_1(u) = \nu_1, \ldots, c_n(u) = \nu_n\} \subseteq \{u \in U : d(u) = \nu_d\}$.

3 Rules of Emergent System

Traditionally, *AI* relied strongly on models of representation and direct perception of the world [14]. It was mostly preoccupied with functional semantics. Emergent systems [4, 7–9, 16] are philosophically and methodologically different from traditional approaches. They replace the traditional high-level control by scheme of functional modularization of behavior generating modules [7]. The characteristics of emergent systems are examined. Instead of high–level computation of behavior, the bottom–up (emergent) self–organization of simpler components produces a variety of behaviors depending on the interaction of the agent with its environment [15, 5]. The structure coupling of the agent and its environment is the source of the behavior, and not just the agent control system alone [14].

The difficulties we face in adopting a definition of the concept of emergence are reminiscent of the complications faced by early Artificial Intelligence (*AI*) researchers in defining intelligence. Researchers are of opinion that the concept of intelligence is unlikely to subside in the foreseeable future, and the same, we believe, holds for emergence, hence, we will consider a system as emergent if:

1. The environment and the system are complex enough to allow the mechanisms of self–organization to occur in the situated interaction of agents in the environment.
2. The system is self–replicating.
3. The system has been constructed by describing local elementary interactions between components.

These conditions are the rules for diagnosing emergence, in other words, accepting that the system has emergent behavior or not. There should be sufficient interaction to have these agents constructing shared semantics by virtue of their situated interactions with the world and each other, which provide a coupling interaction between the system and its environment. Some of the above points deserve further elaboration, or indeed, invite debate. Before treating these issues, we wish to describe emergence in more detail. The emergent system has some characteristics that should be satisfied when we construct it:

- Direct communications only happen with local neighbors.
- Absence of central control.
- The inability of individual participant components to view the state of the system from a global perspective.
- The environment has the sufficient initial richness and structure to allow for embodied emergent classification of that environment/system coupling.

Let us now consider the mathematical meaning of emergence. If n is the size of the system, by $\pi(n)$ we denote the amount of computation required to simulate a system, and arrive at a prediction of the given phenomenon. Let $\omega(n)$ be the amount of computation required to arrive at the result by performing a creative analysis depending on our understanding of the symmetries of the system that help us to deduce the future state whilst circumventing most of the previous-required computation.

Let us consider an idealized means of prediction in that complex system, requiring an amount of computation $\omega_{opt}(n)$ based on perfect understanding [14]. Now, the related measure of understanding is given by

$$\lambda = \frac{\omega_{opt}(n) - \omega(n)}{\pi(n)} \tag{3}$$

So in order to consider a system as an emergence the two conditions are required,

1. $\omega(n) \geq \pi(n)$
2. $\lambda = 0$

Global non-functional properties of a system, including those needed to fulfill mission requirements, performance, often arise through the interactions among components of the system. Global properties that prevail for a system as a whole, but cannot or do not exist within individual components of that system, are called emergent properties [14]. The definition of emergent properties can be formed as follows; E is an emergent property of the system S if and only if

$$E \in V(\nabla S) \text{ and } E \notin V(s_i),$$

where ∇S is the second structure obtained by applying the interaction I to the primitive S, and the observable properties of the primitives $V(S)$, i.e., the property E is emergent iff it is observable on phase ∇S, and not below it.

4 Emergent Rough Sets

We will describe the emergent technique as well as the modification to the standard rough sets theory to construct what is called emergent rough set methodology.

4.1 Objects as the Low-Level Emergent Interaction Component

The main advantage of rough set theory is that it does not need any preliminary or additional information about data like probability in statistics or basic probability assignment in Dempster-Shafer theory, grade of membership or the value of possibility [13]. This means that rough set theory does not need any global information and the global result of the interactions between objects is the target of the method.

Analysis of information systems and, in particular, decision tables, belongs to the most important tasks of Artificial Intelligence. The decision table describes a finite subset of a collection, also referred to as a universe, of objects belonging to different categories called decision classes.

Objects are the lower level of the system and then the interactions among objects are the basis for emergent behavior that can be observed in the system. Let us consider the mathematical formula for interactions between objects in the system. Let a function $\sigma : 2^U \rightarrow [0,1]$ satisfy for all subsets X of universe U,

$$1.\ \sigma(\emptyset) = 0,\ \ \sigma(X) \leq 1 \tag{4}$$

$$2.\ A \subseteq B \subseteq X \text{ implies } \sigma(A) \leq \sigma(B) \tag{5}$$

For $x \notin X$, we let

$$\psi_\sigma(X, x) = \sigma(X \cup \{x\}) - \sigma(X) \tag{6}$$

that denotes the marginal contribution of x to $\sigma(X \cup \{x\})$.

Let us now consider the second order differences, which are called the interaction values,

$$\psi_\sigma(X, \{p_1, p_2\}) = \sigma(X \cup \{p_1, p_2\}) - \sigma(X \cup \{p_1\}) - \sigma(X \cup \{p_2\}) + \sigma(X) \tag{7}$$

where $p_1, p_2 \notin X$. Let us now perform the summation across all subsets of U,

$$\Psi_\sigma(p_1, p_2) = \frac{1}{n} \sum_{X \subseteq U \setminus \{p_1, p_2\}} \psi_\sigma(X, \{p_1, p_2\}) \tag{8}$$

We say that $p_1, p_2 \in U$ show a positive interaction if $\Psi_\sigma(p_1, p_2) > 0$ and they do not interact otherwise.

Consequently each rough set has boundary-line examples, i.e. objects that cannot be with certainty classified as members of the set or its complement. These objects interact with other objects in the set and with the environment (attributes), and the global result from these interactions is the determination of lower and upper approximations.

Table 2. The values of ψ for objects in Table 1.

U	$\|\psi_\sigma(p)\|$
$P1$	0.54
$P2$	0.58
$P3$	0.61
$P4$	0.58
$P5$	0.58
$P6$	0.58
$P7$	0.54
$P8$	0.61
$P9$	0.61
$P10$	0.58

Let take an example of the values of the interaction factor σ between objects as in Table 1 and consider the σ function in the form,

$$\sigma(X) = \frac{|(\overline{X} - \underline{X})|}{|\overline{X}|} \tag{9}$$

Then the values of the function ψ can be as in Table 2.

4.2 Equivalence Relation as Interaction Base

Rough set concept is based on the indiscernibility relation between objects. Suppose we are given a finite set of objects U called the universe and a binary relation I over U called indiscernibility relation. The indiscernibility relation can be an equivalence relation or the tolerance relation. The second is more general than the first one but for simplicity we assume that I is an equivalence relation. The relation I is meant to express the fact that our knowledge about elements of the universe is limited and therefore we are unable, in general, to discern among them. That means that, in general, we are unable to deal with any single element of the universe but only with clusters of similar elements (global view) that from the mathematical point of view are equivalence classes of the indiscernibility relation I.

Let us define what is called a Weighted Information System $WIS = (U, C, W)$, where U is the set of objects, C is the set of attributes, and W is the set of weights related to the set of attributes C connected with each object. The weighted decision table takes the form $WDT = (U, C \cup D, W)$. The values of the set of weights W are determined according to the interactions between objects and between objects and their environment. If object p_1 interacts with object p_2, then weight value for attribute i related to object p_1 is modified according to the formula,

$$w_i = \begin{cases} \alpha, & d_{o_1} = d_{o_2} \\ -\alpha, & d_{o_1} \neq d_{o_2}, \end{cases} \tag{10}$$

where d_{o_i} is the decision value on the object p_i.

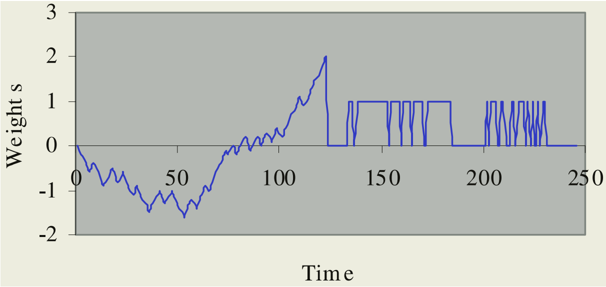


Fig. 1. The modification in the attribute # 4 for random object from Monk data.

The value of α is determined according to similarity of attribute i between these two objects,

$$\alpha = \begin{cases} \tau, & i(p_1) = i(p_2) \\ -\tau, & i(p_1) \neq i(p_2) \end{cases} \tag{11}$$

Weighted information system is a specific information table and, therefore, all concepts of the rough set analysis can be adopted to it. Let us take an example to show how the interaction between objects using the equivalence relation can affect weight values of attributes.

Fig. 1 gives the example to the modification of weight values for one attribute associated with random object taken from Monk data problem. From this figure, it is clear that the weight value can be negative or positive according to the result of interactions among objects.

4.3 Emergent Problem Decomposition of Self-replication

Von Neumann's model [10] of self-replication is a system-theoretic criterion of open-ended evolution. According to Von Neumann's formulation, a symbol system utilized for the construction of self-reproducing systems is not an isolated artifact. Rather, in the context of construction, a symbol system entails a set of available parts. That is, construction blueprints are cast on a symbol system whose primitives are a finite set of parts. A given material symbol system cannot represent everything, only what its primitive parts can construct.

A particular materiality is tied to specific construction building blocks. The richer the parts, the smaller the required descriptions, but also the smaller the number of classifiable categories or constructed morphologies.

Information about object, say x is represented by its attribute value vector. The set of all objects with similar (to attribute value vector) value vectors creates the set $I(x)$. The relation $I(x)$ is in this case an equivalence relation. We assume that to answer the question whether an object x belongs to an object set X, we have to answer a question whether $I(x)$ is in some sense included in X. This measure is introduced by Polkowski and Skowron in their study on rough mereology [13]. Whereas rough mereology firstly applied to distributed information systems, its essential idea is rough inclusion: Rough inclusion focuses on set-inclusion to characterize a hierarchical structure based on a relation between a subset and superset. Thus, application of rough inclusion to capture the relation between classes is equivalent to constructing rough hierarchical structure between classes, which is also closely related to information granulation proposed in [17]. Rough mereology [12, 13] can be regarded as a far reaching generalization of mereology of Lesniewski [13], it does replace the relation of being a part with a hierarchy of relations of being a part in a degree. Hence, we take as a primitive notion a rough inclusion function rather than rough membership function [12].

This idea gives us the interpretation of the statement that the object x is composed from parts $x_1, x_2, \ldots, x_k$. One can use rough inclusion $\mu(x, y) = r$ as the statement: x is an element (subset) of y in degree r. Standard rough inclusion function is defined for two sets $x, y \subseteq U$ by the formula,

$$\mu(x, y) = \frac{|x \cap y|}{|x|} \tag{12}$$

The rough inclusion function $\mu(x, y)$ satisfies the following conditions for any triple x, y and z:

1. $\mu(x, x) = 1$ for any $x \in U$.
2. If $\mu(x, y) = 1$ then $\mu(z, y) \geq \mu(z, x)$.
3. If $\mu(x, y) = 1$ and $\mu(y, x) = 1$ then $\mu(x, z) \geq \mu(y, z)$.
4. There is an object π such that for any object $x: \ \mu(\pi, x) = 1$.
5. For any collection of objects Y there is an object x with the properties:
   a. If $non(\mu(z, n) = \mu(n, z) = 1)$ and $\mu(z, x) = 1$ then there are t such that $non(\mu(t, n) = \mu(n, t) = 1)$ and $\tau \in Y$ such that $\mu(t, z) = \mu(t, \tau) = \mu(\tau, x) = 1$;
   b. If $\tau \in Y$ then $\mu(\tau, x) = 1$;
   c. If y satisfies the above two conditions then $\mu(x, y) = 1$.

Any object x satisfying condition (a) is called a set of objects in Y; if, in addition, x satisfies (b) and (c), then x is called the class of objects in Y. These notions allow for representations of collections of objects as objects.

The formula $x = set(\mu_U)\{x_1, x_2, \ldots, x_k\}$ is interpreted as the statement that the object x is composed from parts $x_1, x_2, \ldots, x_k$. One can interpreted $\mu(x, y) = r$ as: x is an element (subset) of y in degree r; if $y = set(\mu)Y$, then $\mu(x, y) = r$ means that x is a member of the collection Y in degree r.

Rough inclusion can be generated from a given information system IS; for instance, for a given partition $P = \{C_1, \ldots, C_k\}$ of the set of attributes C, and a given set of weights $W = \{w_1, \ldots, w_k\}$, $w_i \in [0, 1]$ for $i = 1, 2, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$ we let:

$$\mu_{P,W}(x, y) = \sum_{i=1}^{k} w_i \cdot \frac{||I_i(x, y)||}{||C_i||} \tag{13}$$

We call it a pre-rough inclusion [13].

Let consider an information system as in Table 3, where the set of objects $U = \{p_1, \ldots, p_6\}$ and the set of attributes $C = \{English, French, German\}$. Table 4 shows values of the initial inclusion when $k = 1$ and $w_1 = 1$.

Table 3. An example information system.

Object	English	French	German
$P1$	Yes	No	No
$P2$	No	No	Yes
$P3$	No	Yes	No
$P4$	Yes	Yes	No

Table 4. Values of rough inclusion for information system in Table 3.

	$P1$	$P2$	$P3$	$P4$
$P1$	1	0.33	0.33	0.66
$P2$	0.33	1	0.33	0.00
$P3$	0.33	0.33	1	0.66
$P4$	0.66	0.00	0.66	1

4.4 Emergence and Discretization

We can talk about two common problems that are inherent to any data mining task: the problem of choosing a suitable data representation and the problem of formulating a goal (or goals) of data modeling [12]. Finding the right representation of data usually involves many steps. First, we have to decide which attributes might be relevant to our problem. Continuous attributes should usually be discretized, i.e., numerical attributes are often rescaled.

When we come to analyze attributes with real values, we must undergo a process called discretization (or quantization), which divides attribute's values into intervals. Such intervals form new values for attribute and, in consequence, allow to reduce the size of attribute's value set.

If $DT = (U, C \cup \{d\})$ is a decision table with large card(V_c) for some $c \in C$, then there is a very low chance that a new object will not match any of these rules; hence, for decision tables with real value attributes, some discretization strategies are built to obtain a higher quality of classification. This problem is intensively studied in [12, 13].

Let $DT = (U, C \cup \{d\})$ be a decision table where $U = \{x_1, x_2, \dots, x_n\}$. We assume $V_c = [l_c, r_c) \subseteq \mathbf{R}$ for any $c \in C$ where $\mathbf{R}$ is the set of real numbers. An arbitrary attribute $c \in C$ defines a sequence $v_1^c < v_2^c < \cdots < v_{n_c}^c$, where $\{v_1^c, v_2^c, \dots, v_{n_c}^c\} = \{c(x) : x \in U\}$. Let P_c be a partition on V_c into subintervals:

$$P_c = \{[b_0^c, b_1^c), [b_1^c, b_2^c), \dots, [b_{k_c}^c, b_{k_c+1}^c)\} \tag{14}$$

for some integer k_c, where $l_c < b_0^c < b_1^c < \cdots < b_{k_c}^c < b_{k_c+1}^c = r_c$ and $V_c = [b_0^c, b_1^c) \cup [b_1^c, b_2^c) \cup \cdots \cup [b_{k_c}^c, b_{k_c+1}^c)$.

Fig. 2 gives an intuitive idea of cuts: "any cut can be treated as a linear $(k-1)$-dimensional surface that divides the affine space R^k into two half-spaces".

Any P_c is uniquely defined by the set of cuts on $V_c : B_c = \{b_1^c, b_2^c \cdots, b_k^c\}$. In the sequel we often identify P_c with the set of cuts on V_c defined by B_c.

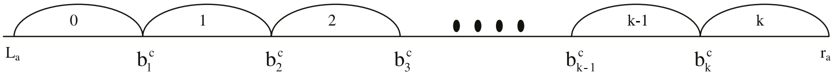


Fig. 2. The partition of the real line defined by the set of cuts $\{b^c_1, \cdots b^c_k\}$.

Then any family $P_c = \{P_c : c \in C\}$ of partitions can be represented in the form $P_c = \bigcup_{c \in C} c \times B_c$. Any $(c, b) \in P_c$ will be also called a cut on V_c.

Any set of cuts $P = \bigcup_{c \in C} P_c$ allows to construct from $DT = (U, C \cup \{d\})$ a new decision table $DT^p = (U, C^p \cup \{d\})$ where $C^p = \{c^p : c \in C\}$ and $c^p(x) = i \Leftrightarrow c(x) \in [b^c_i, b^c_{i+1})$ for any $x \in U$ and $i \in \{0, \cdots, k_c\}$.

Let another set of attributes C' be built from the set C, where C' is defined by $C' = \left\{ c^c_i : c^c_i = \dfrac{v^c_i + v^c_{i+1}}{2}, 1 \leq i \leq n_c - 1 \right\}$.

The set C' is called the initial set of cuts. Any cut $b^c_i \in C'$ can be treated as a Boolean function defined in $U = \{x_1, \ldots, X_n\}$ as follows:

$$b^c_i(x_j) = \begin{cases} 0 & \text{if } c(x_j) < \dfrac{v^c_i + v^c_{i+1}}{2} \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

Our algorithm for discretization is searching for a cut $b \in C'$, which discerns the largest number of pair of objects and moves the cut b from C^t to the resulting set of cuts R using only the local interactions between objects. The procedure is continued until all pairs of objects from U, with different value of decision, are discerned by cuts from the resulting set of cuts R.

The process of discretization for a continuous attribute begins with determining k_c cuts for the values of attribute $c \in C$ according to the Equal Width Method and then construct the initial set of attributes C^t. It involves determining the domain of observed values of an attribute $c \in C$, i.e. $[v^c_{min}, v^c_{max}]$, and dividing this interval into k_c equally sized intervals, where $k_c \in N$ is a parameter. One can compute the interval width: $\delta = \dfrac{v^c_{\max} - v^c_{\min}}{k_c}$ and construct interval boundaries: $b^c_i = v^c_{\min} + i \times \delta$ where $i = 1, \ldots, (k_c - 1)$.

Next step in the process is to allow for objects to interact and adaptive the set of cut values according to these interactions. If object j_1 interacts with object j_2 and they have a different decision values, i.e., $d(j_1) \neq d(j_2)$ then the cut value for attribute c such that j_1 belongs to b^c_h is modified by the value $(\pm(1/(c(j_1) - c(j_2))$. In a similar way, that happens for a cut value for attribute c of object j_2.

Fig. 3 shows the modification values of set of cuts before and after the process of local interaction between objects in the system for one attribute from medical data set (see application section). We observe from the figure that the second cut is deleted when after adaptation it becomes equal to the third cut. So we have when the adaptation is finished four cuts only.

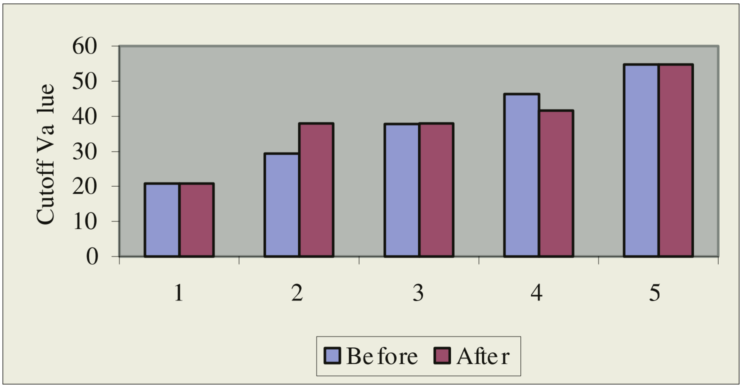


Fig. 3. The adaptive (before and after the interactions between objects) for the set of cuts for one attribute from the medical data.

4.5 Reduction of Information

Standard rough set methods are not always sufficient for extracting laws from decision tables [12]. One of the reasons is that these methods are not taking into account the fact that part of the reduct set is not stable in randomly chosen samples of a given decision table. We propose a method for selection of feature sets relevant for extracting laws from data based on the interactions between objects. These sets of attributes are in most of senses the most stable reduct of a given decision table.

The algorithm for reducing attributes, based on the idea of weighted decision table, takes the form

```
1. Input
     Decision table $DT = (U, C \cup D)$.
2. Output
     Reduct $R$.
3. Process
   a. Convert the decision table $DT$ into the weighted decision table $WDT$
      by adding a distinguished set of weights $W$. Weight values are associated
      with the set of attributes and the interactions between objects (see
      Sect. 4.2).
   b. While (there is no object that does not interact) do
       i. Interact object $i$ with object $j$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, n\}$.
      ii. For ($a = 1$ to $m$)
            Modify the weight value associated with attribute $a$ and object $i$
   c. $R = \emptyset$
   d. For ($i = 1$ to $m$)
      If $\left( \sum_{j=1}^{n} \omega_{ij} > 0 \right)$ then $R = R \cup \{a\}$
```

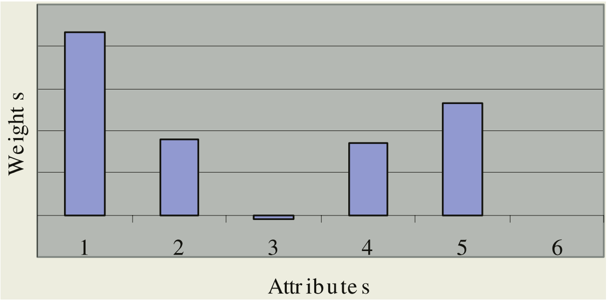


Fig. 4. The values of attribute weights for Monk dataset.

Fig. 4 shows the values of summed weights for attributes for the first Monk dataset (see the section of applications). From this figure, we observe that attribute number one has the largest value while the attribute number three has the lowest value. From Fig. 4, we can calculate the reduct set as {attribute#1, attribute#2, attribute#4, attribute#5}.

4.6 Induction of Decision Rules

The most popular method for classification algorithm construction is based on learning rules from examples. The methods based on calculation of all reducts allow to compute, for a given data, the descriptions of concepts by means of decision rules [6, 17]. Unfortunately, decision rules constructed in this way can often be not appropriate to classify unseen cases. We propose a method of the decision rule generation on the basis of local interactions. We suggest that the rules calculated by means of emergent technique are better predisposed to classify unseen cases.

Decision rules are generated from reducts. So in order to compute decision rules, reducts have to be computed first. When a set of reducts has been computed then it is necessary to decide how to compute the set of decision rules. We discuss methods based on the decision rule set calculation for any reduct. We have implemented several methods. Let us consider two methods. Using the first method we generate decision rules for the reduct R in the following way: for any object, we take the value vector of conditional attributes from R and the corresponding decision about the object. Unfortunately, the decision rules generated by this method have bad performance, because the number of objects supporting such rules is usually very small.

By applying the second method, we generate decision rules with minimal number of descriptors for the table consisting of the object set equal to the reduct domain of R, the conditional attribute set equal to R.

4.7 Emergent Phenomena of Classification Process

Emergent phenomena are those for which the amount of computation necessary for prediction from an optimal set of rules, classifications and analysis,

Table 5. The weight values for the set of rules induced from Monk data.

Rule	Weight value
IF Sex=F AND Clinical Stage=B ⇒ Infection=No	0.6
IF Sex=F AND Clinical Stage=T ⇒ Infection=Yes	0.2
IF Sex=M AND Clinical Stage=B ⇒ Infection=No	0.0

even derived from an idealized perfect understanding, can never improve upon the amount of computation necessary to simulate the system directly from our knowledge of the rules and its interactions.

The behavior of a situated object in an environment is, in this sense, the result of how the object classifies the environment. Thus, behavior is itself an emergent property of the low level dynamics that implements the object. The classification process is emergent because it is the result of the local interactions of the basic components of the system and does not stem from a global controller.

Classification process of rough set methodology represents emergent computation in the strict sense. What emerges from the process is a highly adapted rule-set that interacts with its environment to solve problems. This emergence is the result of interactions between quasi-independent entities (classifications) via the mechanisms specified in the bucket brigade, and between the classifiers and their environment.

Another interesting form of emergence occurs when the logical language for knowledge discovery in databases resulting from the rough set process includes a decision rule. When the conditional semantics of the form IF ⟨condition⟩ THEN ⟨decision⟩ is available, the otherwise haploid structure creates a task dependent representation with some interesting dominance characteristics. For each rule, the condition part is first evaluated and based on its result the decision part is taken. Since the condition part is dependent on the state of the environment, then the decision part of the rule emerges dynamically and then it is determined by the context of the environment. This permits an environment specific form of dominance.

By giving a weight value for each rule, we can sort the set of rules and classify any object to the first rule in the list that match it first. The weight value for rule r that is derived from the weighted decision table $WDT = \{U, C \cup \{d\}, W\}$, takes the form:

$$weight(r) = \sum_{p\_match\_r} \frac{\sum\limits_{i \in r} W_{pi}}{|r|} \tag{16}$$

where W_{pi} is the weight value of attribute i that exists in the condition part of the decision rule r, and object p matches the rule r, and |r| is the number of condition attributes that are contained in rule r.

Table 5 gives the weight values for set of rules that is induced from the decision table in Table 1.

5 Guidelines to Increase Emergence in Rough Sets

In this section we will present some guidelines for directed emergent phenomena in rough set theory. These guidelines should be taken in mind when we construct a system based on rough sets that contains emergent phenomena.

Brute force methods [1], i.e., ones that blindly attack the problem, typically provide little benefit, although they may provide a short-term benefit for simple problems. Such strong coercion of the rough set method dynamics only compromises the natural dynamics of the process. In fact, directly attacking the problem often removes the ability for features to emerge from the dynamics altogether. The desired properties should be coaxed out naturally by the dynamics of the computation. Thus the solution should come from within the task. The algorithm job is only to reinforce the progression toward improvement rather than restrict the method that accomplishes it.

For most problems, there is no single rule matching the objects that is uniformly proficient across all possible decisions. In standard rough sets, the static rules engineered knowledge ends up being either too general or too specific to cover many objects. Opportunistic method like emergent latches on to whatever provides a benefit in the current problem-solving situation. Each rule may produce a completely distinct classification since the features that emerge during processing are dependent on the points in the problem space visited by the algorithm. Generality of features that emerge are often minimal, but they are pertinent to the specific problem or object begins attacked. The opportunism of emergence and the diversity of the resulting structures may enlighten us as to what our biases are and how they limit our ability to solve problems.

6 Applications

To verify the usefulness of the presented methodology, computational experiments have been performed. We report results of experiments on the Monk problems data and medical dataset.

Each dataset was divided into a training set and testing set, and the computations of rules have been done only on training data. The results of computations of rules were applied to the classification of objects from the testing dataset. To evaluate the classification process, we use a measure called classifier's error rate, which is defined by the ratio of the number of errors to the number of all cases.

6.1 Monk Problem Dataset

The attribute information in each monk problem is
Attribute#1: {1, 2, and 3}
Attribute#2: {1, 2, and 3}
Attribute#3: {1, 2}
Attribute#4: {1, 2, and 3}
Attribute#5: {1, 2, 3, and 4}
Attribute#6: {1, 2}

Table 6. Results from the first Monk problem data.

	Monk1	**Monk2**	**Monk3**
C4.5	00.00 %	34.72%	3.70%
Rough Set	11.32 %	22.91%	5.13%
Proposed Method	00.00 %	30.09%	2.77%

Thus, the six attributes span a space of $432 = 3 \times 3 \times 2 \times 3 \times 4 \times 2$ examples for monk's data.

The First Monk Problem

It is to learn a concept description that can be formulated as a simple disjunctive normal form. The target concept can be simply characterized by an expression: [Attribute#2=1] OR [Attribute#4= Attribute#5] $\diamondsuit$ M1, which can be interpreted as: "if for an unknown entity, Attribute#2 takes value 1, or Attribute#4 and Attribute#5 take the same value, then regardless of the values of other attributes classify the entity as an instance of concept M1". The set of training set is 124 training examples (62 positive and 62 negative). The set of testing examples includes all possible 432 cases (216 positive and 216 negative).

The Second Monk Problem

It relates to the following concept "if exactly two of six given attributes take their first value for the given entity, then classify the entity as an instance of class M2". The training set contains 169 examples (105 positive and 64 negative), and the test data is 432 cases (190 positive and 242 negative).

The Third Monk Problem

It is to learn a concept which is: IF [Attribute#2 $\neq$ 4] and [Attribute#5 =1 or 2] or [Attribute#1=1] and [Attribute#2=3] $\diamondsuit$ M3. The training set contains 122 examples (62 positive and 60 negative), and test data is 432 cases (204 positive and 228 negative). It has 5% additional noise (misclassifications) in the training set.

Table 6 gives the information about results of our experiments with Monk problems data. The set of rules that is produced by our method is the best result with respect to third Monk dataset. For second Monk data, our new method gives result better than C4.5 but not better than standard rough set method. For the first Monk dataset, our proposed method and C4.5 give the same result which is better than standard rough set method.

6.2 Medical Dataset (Biopsy Dataset)

We report results of experiments on the Medical datasets (Biopsy dataset). The data was obtained from the Department of Urology and Pathology, Kitasato University School of Medicine, Japan. More details about the data were given in [2, 3].

Table 7. The attribute description of the medical dataset (Biopsy dataset).

Attribute Name	Description
Age	The age of patients
Total PSA	Prostate specific antigen.
Free PSA	Free prostate specific antigen
Free Total ratio	Free PSA / Total PSA
PSA density	Total PSA / Gland Vol.
PSATZ	Total PSA / Transitional zone volume in prostate
Gland vol.	Prostate size
TZ vol.	Transitional zone in prostate
DRE	Digital rectal examination

Table 8. The result of classification error for medical dataset.

Proposed method	Neural Networks	Standard Rough Set
0.146	0.159	0.162

The description of the attributes is shown in Table 7. The decision attribute is the pathological result; 0 means no cancer, and 1 means has cancer.

The present method shows the highest accuracy for prediction of pathological stage after radical surgery and biopsy result. As new strategy, the model is essential for diagnosis and treatment selection of prostate cancer.

As the result from our new approach with the medical data, we observe very good rules like:

IF (2.16<Total PSA<9.81) **AND** (0.0693<Free Total ratio<0.53) **AND** (0.07<PSA density<0.54) **AND** (0.11<PSA TZ density<2.35) **AND** (16.619<Gland vol<141.4) **AND** (3.5<TZ vol<87.7) **AND** (1.5<DRE<2.0) **THEN** (Patho=0).

This rule is true for 148 cases from 246 is the size of data with accuracy=0.840909 and rule coverage=0.712551. As well as the rule:

IF (2.16<Total PSA<9.81) **AND** (0.0693<Free Total ratio<0.53) **AND** (0.07<PSA density<0.54) **AND** (0.11<PSA TZ density<2.35) **AND** (16.619<Gland vol<141.4) **AND** (3.5<TZ vol<87.7) **AND** (3.5<DRE<4.0) **THEN** (Patho=0)

Which has support as 13 cases with rule accuracy=0.565217 and rule coverage= 0.093117.

There exist a set of rules that is induced from C4.5 method, but there does not exist any rule can cover many cases as first rule.

Table 8 shows the classification error for our new method and compares it with artificial neural networks and standard rough set method. We get the set of cuts for neural network and standard rough sets from the medical domain. From Table 8, we observe that our method is the best result than neural networks model and standard rough set method.

Table 9. The cuts values derived from the proposed method for medical dataset.

Attribute	Cut 1 values	Cut 2 value	Cut 3 value	Cut 4 value	Cut 5 value
Age	39-58	58-60	60-68	68-91	———
Total PSA	2.05-2.11	2.11-2.1669	2.1669-9.81	9.81-9.93	———
Free PSA	0.21-0.4	0.4-1.3029	1.3029-5.21	———-	———
Free Total ratio	0.04-0.05	0.05-0.06	0.06-0.0693	0.0693-0.53	0.53-0.71
PSA density	0.04-0.06	0.06-0.07	0.07-0.54	———	———
PSA TZ density	0.07-0.077	0.077-0.11	0.11-2.35	———	———
Gland vol	8.2-15.4	15.4-15.6	15.6-16.619	16.619-141.4	———
TZ vol	1.8-3.5	3.5-87.7	———	———	———
DRE	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0

Only C4.5 method can make discretization for continuous attributes same as our proposed method, but C4.5 discretization is binary divisions for the possible values of the attribute. We show the set of cuts that is derived from our algorithm as in Table 9. From this table, we observe that the set of cuts is better from the view of logic than binary division of C4.5 method.

7 Conclusions

The objective of this paper was to setup a methodology for the emergent rough set design which in particular terms means that we were supposed to propose a fully emergent system for implementing basic rough set theory notions. The research results will have important impact on development of new methods for KDD, in particular for development of algorithmic methods for pattern extraction from data. We are convinced that progress in the above areas is of the utmost importance for creating new methods, algorithms, software as well as hardware systems which prove the applicability of rough set techniques to challenging problems of data mining, knowledge discovery and other important areas of applications.

In this approach, we define the term emergence and give the rules that must be satisfied to the system to be emergence. Next, we present some new techniques for rough set methodology that depend on the emergent computation. The global results of these techniques depend only on local interactions between objects. In this work we have discussed the discretization problem with regard to the discernibility relation of objects defined by cuts.

In the practical parts of our study, we use the Monk data problem and medical dataset. We need to continue this direction of study toward complete emergent system that depends on rough sets theory. We have now an interesting open problem for future research as a continuous decision problem. We have considered decision tables with discrete decision. However, for some problems (e.g., control problems) one should take into account that the decision is continuous. New strategies for discretization of attributes in decision tables with continuous decision are needed.

References

1. Angeline, Peter J. Genetic programming and emergent intelligence, in: K. E. Kinnear, Jr.(ed.), *Advances in Genetic Programming*, MIT Press, 1994
2. Egawa, S., Suyama, K., Yoichi, A., Matsumoto, K.,Tsukayama, C., Kuwao, S., and Baba, S., A study of pretreatment nomograms to predict pathological stage and biochemical recurrence After radical prostatectomy for clinically resectable prostate cancer in Japanese men, *Japan Journal of Clinical Oncology* **31**(2) (2001) 74–81
3. Egawa, S., Suyama, K., Matsumoto, K., et. al, Improved predictability of extracapsular extension and seminal vesicle involvement based on clinical and biopsy findings in prostate cancer in Japanese men, *Urology* **52**(3) (1998) 433–40
4. Gotts, N., Emergent phenomena in large sparse random arrays of Conway's 'Game of Life', *International Journal of System Science* **31**(7) (2000) 873–894
5. Hanson, J., Crutchfield, J., Computational mechanics of cellular automata: An example, *Physica* **D103** (1997) 169–189
6. Hassan, Y., Tazaki, E., Knowledge discovery using rough set combined with genetic programming, *Proceeding of RSTGC'2001 International Conference* **5**(1) (2001) 75–79
7. Hassan, Y., Tazaki, E., Emergence in combined system structure of rough set theory and genetic programming, in: N. Baba, L. C. Jain, R. D. Howlett (Eds.), *Proceeding of Knowledge-based Intelligent Information Engineering Systems & Allied Technologies: International Conference KES' 2001, Part 2*, IOS press, 2001, pp.1458–1462
8. Hassan, Y., Tazaki, E., Adaptive behavior in cellular automata using rough set theory, *Applied Artificial Intelligence Journal* **17**(2) (2003) 1–21
9. Hassan, Y., Tazaki, E., Emergence computation using new model of cellular automata, *Applied Artificial Intelligence Journal*, **17**(1) (2003) 39–69
10. Olson, R., Sequeira, R., Emergent computation and the modeling and management of ecological systems, *Computers and Electronics in Agriculture* **12** (1995) 183–209
11. Pawlak, Z., Rough sets, *International Journal of Computer and Information Science* **11**(5) (1982) 341–356
12. Polkowski, L., Skowron, A. (Eds.), *Rough Sets in Knowledge Discovery*, Physica-Verlag, Heidelberg, 1998
13. Polkowski, L., Tsumoto, S., Lin, Y., (Eds), *Rough Set Methods and Applications*, Physica-Verlag, Heidelberg, 2000
14. Rocha, L., Joslyn, C., Simulations of embodied evolving semiosis: Emergent semantics in artificial environments, *The Proceedings of the Conference on Virtual Worlds and Simulation*, 1998, pp.233–238
15. Sipper, M., Studying artificial life using a simple, general cellular model, *Artificial Life Journal* **2**(1) (1995) 1–35
16. Thornton, C., Brave mobots use representation: Emergence of representation in fight-or-flight learning, *Minds and Machines* **7** (1997) 475–494
17. Ziarko, W. (Ed.), *Rough Sets, Fuzzy Sets, and Knowledge Discovery*, Springer-Verlag, Berlin, 1994

Author Index