STUDIA UNIVERSITATIS
BABEŞ-BOLYAI

INFORMATICA

1/2007

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

THE FIRST INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING PRINCIPLES AND TECHNIQUES (KEPT 2007)

DOINA TĂTAR, HORIA F. POP, MILITON FRENŢIU, AND DUMITRU DUMITRESCU

1. Introduction

The Faculty of Mathematics and Computer Science of the Babeş-Bolyai University in Cluj has organised during June 6-8, 2007, the First International Conference on Knowledge Engineering Principles and Techniques (KEPT 2007), a beginning of what we would like to be a permanent series of events on theoretical foundations and real-world applications of knowledge engineering. This conference, organised on the platform of Knowledge Engineering, has been a forum for intellectual, academic, scientific and industrial debate to promote research and knowledge in this key area, and to facilitate interdisciplinary and multidisciplinary approaches, more and more necessary and useful today.

The conference was honoured by leading class keynote lecturers, to present their invited lectures in two plenary sessions: Prof. Bruno Buchberger (Johannes Kepler University Linz), with a lecture on "Algorithm Synthesis by Lazy Thinking: A Case Study in Mathematical Knowledge Engineering", Prof. Diana Inkpen (University of Ottawa, Canada), with a lecture on "Semantic Similarity Knowledge and its Applications", Prof. Alain Lecomte (University Paris 8, France), with a lecture on "Some Representation Structures for Computational Linguistics", Prof. Rada Mihalcea (University of North Texas, USA), with a lecture on "Using Wikipedia for Automatic Word Sense Disambiguation", and Dr. Constantin Orasan (University of Wolverhampton, UK), with a lecture on "The Role of Linguistic Information for Shallow Language Processing".

The organisation of this conference reflects three major areas of concern: Natural Language Processing, Artificial Intelligence, and Software Engineering. Oral presentations of 41 regular papers are organized in 12 dedicated sessions, planned with sufficient length to encourage the direct dialogue and exchange of ideas among researchers.

Received by the editors: July 12, 2007.

Knowledge engineering refers to the building, maintaining, and development of knowledge-based systems. It has a great deal in common with software engineering, and is related to many computer science domains such as artificial intelligence, databases, data mining, expert systems, decision support systems and geographic information systems. Knowledge engineering is also related to mathematical logic, as well as strongly involved in cognitive science and socio-cognitive engineering where the knowledge is produced by socio-cognitive aggregates (mainly humans) and is structured according to our understanding of how human reasoning and logic works.

Since the mid-1980s, knowledge engineers have developed a number of principles, methods and tools that have considerably improved the process of knowledge acquisition and ordering. Some of the key issues follow: there are different types of knowledge, and the right approach and technique should be used for the knowledge required; there are different types of experts and expertise, and methods should be chosen appropriately; there are different ways of representing knowledge, which can aid knowledge acquisition, validation and re-use; there are different ways of using knowledge, and the acquisition process can be goal-oriented; there are structured methods to increase the acquisition efficiency.

2. Opening lectures

KEPT 2007 conference has been opened with three very actual and interesting lectures. First is called "Semantic Similarity Knowledge" and was done by Dr. Diana Inkpen from University of Ottawa, Canada. It present several methods for computing the similarity of two words, following two directions: dictionary-based methods that use WordNet, Roget's thesaurus, or other resources, and corpus-based methods that use frequencies of co-occurrence in corpora (cosine method, latent semantic indexing, mutual information, etc). Also, several applications of word similarity knowledge are presented: detecting words that do not fit into their context (real-word error correction), detecting speech recognition errors, solving TOEFL-style synonym questions, and synonym choice in context (for writing aid tools).

The second lecture is "The Role of Linguistic Information for shallow Language Processing" and the author is Dr. Constantin Orasan from School of Humanities, Languages and Social Sciences, University of Wolverhampton. The lecture argues that the advantage of shallow methods in comparison to deep processing methods is that they do not require the building of elaborate representations of the text to be processed or to perform reasoning on this data, and as a result they can be more easily implemented. The talk shows how a shallow method for automatic

summarization can improve its performance by adding different types of linguistic information.

The final talk was that presented by Dr. Rada Mihalcea, from the University of North Texas with the name "Using Wikipedia for automatic Word Sense Disambiguation". Starting with the hyperlinks available in Wikipedia, the author shows how one can generate sense annotated corpora that can be used for building accurate and robust sense classifiers.

3. The Natural Language Processing section

To continue with the Natural Language Processing section, the following papers have been presented, accordingly with the order of presentation.

It is known that he task of function tagging involves labeling certain nodes in an input parse tree with a set of functional marks such as logical subject, predicate, etc. The paper "Large Scale Experiments with Function Tagging" shows haw two Decision Trees based approaches to the task of function tagging outperform baseline approaches when the most frequent tag is assigned.

The second paper in NLP section was "A chain dictionary method for Word Sense Disambiguation and applications". The approach of WSD, which is one of the most important open problems in NLP, uses the lexical base WordNet for a new algorithm originated in Lesk's, namely "chain algorithm for disambiguation" of all words (CHAD). Some experiments and evaluations with CHAD for Semcor and Senseval 1 corpora are described, which prove the performance of the algorithm. Conclusions of using the CHAD for machine translation (here from Romanian language to English) and for text entailment verification are discussed.

The following paper, "Text entailment verification with text similarities", presents a method based on lexical resolution and supposes the word sense disambiguation of the two texts (text and hypothesis). The method also relies on a recent directional measure of semantic similarity between two texts and is applied to the dataset of PASCAL RTE-2.

The paper "Syntagma Processing for incomplete answers" refers to the syntagma as an incomplete answer that resumes to a phrase not to a grammatically correct sentence. SPEL (Syntactic Parser for English Language) system is introduced as an method that is able to reconstruct the answers from the given syntagma and the initial question, without affecting the semantic information given by the answer. The paper proposes a solution to incomplete answer processing, answers that are very frequent in a usual communication scenario based upon question-answer pattern.

Named "A Text Analysis Based Approach for the Compliance between the Specification and the Software Product", the next paper proposes a new approach in evaluating the compliance between software documentation (expressed on natural language) and the final software product. The authors define two evaluation measures and present some case studies.

"Text Categorization experiments using Wikipedia" shows how to use Wikipedia articles to give word distributional representation for documents. Since the word-distributional representation causes dimensionality increase, dimensionality reduction is needed to make the problem computationally tractable. The authors use in this respect a method known as latent semantic indexing (LSI) and combine this with the processing of the training corpus. The results of experimenting with the method on real-world dataset is presented.

The paper "The 'Integral' Model of Language Functioning (E. Coseriu)" explores the framework of Coseriu's "integral linguistics", focusing mainly on the three planes of language and their corresponding "linguistics" - the three directions in language investigation that Coseriu postulated. It is argued that, in the panorama of contemporary approaches to language, Coseriu's integral linguistics offers one of the most comprehensive and finely articulated frameworks for investigating the functioning of language in a dynamic perspective. This paper relies on experiences of a pure linguistic team and could seed a further collaboration with our NLP researchers.

"Enhancing the Invisible Web" is the title of next presented paper. This article describes the architecture of an Invisible-Web Extractor, whose primal goal is to enhance the value of the hidden Web data. The author considers three main issues of the tool: how to access the Invisible Web information, how to extract information from the gathered data and how to create new knowledge from it.

In "Chain Algorithm used for Part of Speech Recognition" the author shows how CHAD algorithm (see above) can be used to identify the part of speech of words from a text written in a single language.

Natural Language Generation (NLG) and Foreign Language Writing Aid (FLWA) are two important tasks of Natural Language Processing (NLP), which deal with obtaining natural language from a machine representation system and building computer programs that assists a non-native language user in writing decently in a target language, respectively. The paper "Natural Language Generation - Applications for Romanian Language" uses an affix grammar to construct the Romanian language grammar and a semantic which gives us information about the words we use to build a sentence. It shows how one can construct, starting from a set of words, correct sentences from syntactic and semantic point of view.

The short enumeration of the conference papers presented at KEPT2007 gives an image of the diversity and the depth of tackled problems. We can say that NLP section of Kept2007 was a success and we hope for a continue development in the further editions.

The second day started with the wonderful conference of Prof. Bruno Buchberger (Johannes Kepler University Linz), with a lecture on "Algorithm Synthesis by Lazy Thinking: A Case Study in Mathematical Knowledge Engineering". The lecture stated that Mathematical Knowledge Engineering equals the Algorithm-supported Mathematical Theory Exploration: invent axioms and definitions for notions (functions and predicates), invent and prove theorems about notions, invent problems about notions. The talk focused to the author's method of invention (synthesis) of correct algorithms from problem specifications by systematic reasoning by the "lazy thinking" method. The method is implemented in the Theorema system on top of Mathematica. An example of non-trivial problem is the construction of Grbner bases, which the author presented in details.

4. The Artificial Intelligence section

The Artificial Intelligence section of the conference is focused on Computational Intelligence techniques. The main topics concern Evolutionary Computing, probabilistic neural networks, multi-agent systems and complex networks. Some papers in this section introduce new evolutionary models and concepts applying them for solving complex optimization problems.

A new selection operator based on the family line of each individual is proposed and tested for solving TSP problems (paper "Collaborative Selection for Evolutionary Algorithms").

A novel trajectory based technique intended to solve complex problems via hierarchical decomposition is presented in the paper 'Exact Model Building in Hierarchical Complex Systems'. The potential of the method resides on the ability to aggressively explore the building block space. Two papers deal with the application of genetic chromodynamics metaheuristic in data mining, training and job scheduling. The paper 'Multi-agent Distributed Computing' investigates the potential of intelligent agents to support distributed collaboration environments. A new multi-agent knowledge management and support system in designed and evaluated.

The potential of hybridization between multi-agent systems and nature-inspired metaheuristics such as Ant Colony Systems is explored in the papers 'Stigmergic Agent Systems for solving NP-hard Problems' and 'Sensitive Ant Systems in

Combinatorial Optimization'. Numerical experiments indicate that the emerging models are very promising for solving search problems.

An interesting approach for generating blood networks is described in the paper 'Simulating Microcapillary Networks using Random Graphs'. Solving significant NP-hard problems using evolutionary algorithms and hybrid techniques is an important issue of Computational Intelligence that has been addressed by four papers. The differential evolution technique is used for unsupervised clustering of documents. The paper 'An Evolutionary Model for Solving Multiplayer Noncooperative Games' proposes a technique for detecting multiple Nash equilibria. The method can be extended to deal with cooperative games.

A hybrid technique for parameter setting in probabilistic neural networks is applied for hepatic cancer diagnosis. A supervised version of learning vector quantization induces a neural network for mining a toxin database.

The papers in this section form an important contribution to the field of Natural Computing and address some significant practical applications. New powerful computational models for search and optimization are proposed and some interesting hybridization techniques with a great potential are investigated.

5. The Software Engineering section

Software Engineering refers to specification, design, coding, verification, and maintenance of software. It is connected to knowledge engineering since it implies transforming the requirements of the clients (i.e. knowledge from their domains) into specifications of corresponding software product. In other words, software engineering is dealing with the transformation of knowledge into software. That's why the third section of KEPT2007 was Software Engineering , with its various subdomains.

In the paper "On Software Attributes Relationship Using a New Fuzzy C-Bipartitioning Method", the authors introduce a new data analysis method, the fuzzy bipartitioning method, and use this method to study the dependence between software attributes.

The second paper, "Some Formal Approaches for Dynamic Life Session Management", introduces three formal approaches for determining, establishing, and maintaining the lifetime of a HTTP session.

The third paper, "Management of Web Pages Using XML Documents" describes a method of automatic management of a WEB site, by memorizing in a database the information sources in different pages.

"A View on Fault Tolerant Techniques Applied for Mediogrid", analyses the characteristics of fault tolerance for grid systems. Some directions for enhancing the fault tolerance of the MedioGrid are suggested.

Then "A New Graph-Based Approach in Aspect Mining" presents a new graph-based approach in aspect mining. The cross cutting concerns is viewed as a search problem in a graph, and an algorithm, GRAM, is given to solve the problem.

Next paper, "Introducing Data-Distributions into PowerList Theory", introduces data-distributions into PowerList theory in order to reconcile abstraction of this theory with performance.

The notion of internally and externally stable set for the G-complex of multi-ary relations are defined in "The Stable Sets of a G-complex of Multi-ary Relations and its Applications" Also, some properties of these sets are proven.

The paper "Multi-Agent System for Competence Modeling" gives a model for competencies of people. This model is useful to the universities, and companies, to study the competencies.

Then, "Data Verification in ETL Processes" uses metrics on partitions based on Shannon entropy in the verification of consistency of data loaded into the data warehouse by ETL processes.

Next paper, "Data Predictions using Neural Networks", proposes the Artificial Neural Networks as trainable tools that attempt to mimic information processing patterns in the brain, and use them for data analysis and prediction.

The paper "Evaluating Dynamic Client-Driven Adaptation Decision Support in Multimedia Proxy-Caches" evaluates the use of dynamic client-driven adaptation decision support in multimedia proxy-caches through the use of the Adaptation-awareMultimedia Streaming Protocol.

In "Automated Proof of Geometry Theorems Involving Order Relation in the Frame of the Theorema Project" the author proposes a method that combines the area method for computing geometric quantities and the Cylindrical Algebraic Decomposition method, in order to prove geometry theorems. An implementation of this method as part of Geometry Prover in the frame of Theorema project is done.

The following paper, "A Hierarchical Clustering Algorithm for Software Design Improvement", presents a new hierarchical clustering algorithm that can be used for improving software system design. This approach may be used to assist software engineers in refactoring software systems.

In the paper "Metrics-Based Selection of a Component Assembly" some software metrics are used to select the specified components required by the design of a system.

The authors of the paper "Architecting and Specifying a Software Component using UML" suggest a component-based architecture for LCD Wallet Travelling Clock case study.

In the next paper, "A TSpaces Based Framework for Parallel-Distributed Applications", a framework to deploy and execute parallel-distributed applications is suggested.

Finally, in the paper N. Magariu, Applying Transition Diagram Systems in Development of Information Systems Dynamic Projects a model of complex software development based on the usage of transition diagrams system.

6. Conclusions

The First International Conference on Knowledge Engineering Principles and Techniques (KEPT 2007) was an exciting and useful experience and exchange of knowledge for our department. The possibility to communicate our most recent studies, the commitment with the results of others colleagues, the emulation of new ideas and research, all these mean a great gain of experience in our professional life. We hope that the next edition of KEPT (in 2009) will be even more successful and more enthusiastic than this one.

So, let us see you at KEPT 2009!

E-mail address: dtatar@cs.ubbcluj.ro
E-mail address: hfpop@cs.ubbcluj.ro
E-mail address: mfrentiu@cs.ubbcluj.ro
E-mail address: ddumitr@cs.ubbcluj.ro

Babeş-Bolyai University, Department of Computer Science, 1, M. Kogălniceanu St., 400084 Cluj-Napoca, Romania

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

SEMANTIC SIMILARITY KNOWLEDGE AND ITS APPLICATIONS

DIANA INKPEN

Abstract. Semantic relatedness refers to the degree to which two concepts or words are related. Humans are able to easily judge if a pair of words are related in some way. For example, most people would agree that *apple* and *orange* are more related than are *apple* and *toothbrush*. Semantic similarity is a subset of semantic relatedness. In this article we describe several methods for computing the similarity of two words, following two directions: dictionary-based methods that use WordNet, Roget's thesaurus, or other resources; and corpus-based methods that use frequencies of co-occurrence in corpora (cosine method, latent semantic indexing, mutual information, etc). Then, we present results for several applications of word similarity knowledge: solving TOEFL-style synonym questions, detecting words that do not fit into their context in order to detect speech recognition errors, and synonym choice in context, for writing aid tools. We also present a method for computing the similarity of two short texts, based on the similarities of their words. Applications of text similarity knowledge include: designing exercises for second language-learning, acquisition of domain-specific corpora, information retrieval, and text categorization. Before concluding, we briefly describe cross-language extensions of the methods for similarity of words and texts.

1. Methods for Word Similarity

Semantic relatedness refers to the degree to which two concepts or words are related (or not) whereas semantic similarity is a special case or a subset of semantic relatedness. Humans are able to easily judge if a pair of words are related in some way. For example, most would agree that *apple* and *orange* are more related than are *apple* and *toothbrush*. Budanitsky and Hirst [4] point out that semantic similarity is used when similar entities such as *apple* and *orange* or *table* and *furniture* are compared. These entities are close to each other in an *is-a* hierarchy. For example, *apple* and *orange* are hyponyms of *fruit* and *table* is a hyponym of *furniture*. However, even dissimilar entities may be semantically related, for example, *glass* and *water*, *tree* and *shade*, or *gym* and *weights*. In this

Received by the editors: June 19, 2007.

2000 *Mathematics Subject Classification.* 91F20, 68T35.

1998 *CR Categories and Descriptors.* [I.2.7]: Natural Language Processing; [I.2.4]: Knowledge Representation Formalisms and Methods.

case the two entities are intrinsically not similar, but are related by some relationship. Sometimes this relationship may be one of the classical relationships such as meronymy (*is part of*) as in *computer – keyboard* or a non-classical one as in *glass – water*, *tree – shade* and *gym – weights*. Thus two entities are semantically related if they are semantically similar (close together in the *is-a* hierarchy) or share any other classical or non-classical relationships. Measures of the semantic similarity of words have been used for a long time in applications in natural language processing and related areas, such as the automatic creation of thesauri [6], [18], [17], automatic indexing, text annotation and summarization [20], text classification, word sense disambiguation [15], [17], information extraction and retrieval [3], [30], lexical selection, automatic correction of word errors in text [4], and discovering word senses directly from text [23]. A word similarity measure was also used for language modeling by grouping similar words into classes [1].

There are two types of methods for computing the similarity of two words: dictionary-based methods (using WordNet, Roget's thesaurus, or other resources) and corpus-based methods (using statistics). There are also a few hybrid methods that combine the two types.

Most of the dictionary-based methods compute path length in WordNet, in various ways. A short path means a high similarity. For example, using the WordNet entries for the words *apple* and *orange* the path length is 3:

```
apple  (sense 1)
   => edible fruit
       => produce, green goods, green groceries, garden truck
           => food
               => solid
                   => substance, matter
                       => object, physical object
                           => entity
orange (sense 1)
   => citrus, citrus fruit
       => edible fruit
           => produce, green goods, green groceries, garden truck
              => food
                => solid
                   => substance, matter
                       => object, physical object
                           => entity
```

The WordNet::Similarity Software Package[1] implements several WordNet-based similarity measures: Leacock & Chodorow (1998) [14], Jiang & Conrath (1997) [12], Resnik (1995) [25], Lin (1998) [18], Hirst & St-Onge (1998) [7], Wu & Palmer

[1]http://www.d.umn.edu/~tpederse/similarity.html

(1994) [28], extended gloss overlap, Banerjee & Pedersen (2003) [2], and context vectors, Patwardhan (2003) [24].

If the two words have multiple senses, the similarity between them, out of context, is the maximum similarity between any of the senses of the two words. Three of the above methods are hybrid (Jiang & Conrath (1997) [12], Resnik (1995) [25], Lin (1998) [18]), they use frequency counts for word senses from Semcor, which is a small corpus, annotated with WordNet senses.

Other resources that can be used are thesauri, such as Roget's Thesaurus. For example, the words *apple* and *orange* are in the same paragraph in Roget, but not in the same semicolon group:

```
301 FOOD
n.
fruit, soft fruit, berry, gooseberry, strawberry, raspberry,
loganberry, blackberry, tayberry, bilberry, mulberry;
currant, redcurrant, blackcurrant, whitecurrant;
stone fruit, apricot, peach, nectarine, plum, greengage, damson, cherry;
apple, crab apple, pippin, russet, pear;
citrus fruit, orange, grapefruit, pomelo, lemon, lime, tangerine,
clementine, mandarin;
banana, pineapple, grape;
rhubarb;
date, fig;
```

A similarity measure using Roget's thesaurus [11] computes the distance between the words by exploiting the structure of the thesaurus (path length):

- Length 0: same semicolon group. Example: *journey's end – terminus*
- Length 2: same paragraph. *devotion – abnormal affection*
- Length 4: same part of speech. *popular misconception – glaring error*
- Length 6: same head. *individual – lonely*
- Length 8: same head group. *finance – apply for a loan*
- Length 10: same sub-section. *life expectancy – herbalize*
- Length 12: same section. *Creirwy (love) – inspired*
- Length 14: same class. *translucid – blind eye*
- Length 16: in the thesaurus. *nag – like greased lightning*

Corpus-based methods use frequencies of co-occurrence in corpora. They range from the classic vector-space model (cosine, overlap coefficient, etc.) and latent semantic analysis, to probabilistic methods such as information radius and mutual information.

Examples of large corpora are the British National Corpus (BNC) (100 million words), the TREC data mainly newspaper text, the Waterloo Multitext corpus of webpages (one terabyte), the LDC English Gigabyte corpus, and the Web itself.

Examples of corpus-based measures are[2]: Cosine, Jaccard coefficient, Dice coefficient, Overlap coefficient, L1 distance (city block distance), Euclidean distance (L2 distance), Information Radius (Jensen-Shannon divergence), Skew divergence, and Lin's Dependency-based Similarity Measure[3].

The classic vector space model represents all the words as vectors in an high-dimensional space where the dimensions are the documents (we build a matrix of words by documents). The cosine between two vectors gives the similarity of two terms.

Latent Semantic Analysis (LSA) [4] [13] produces a reduced words by documents matrix, which has fewer dimensions corresponding to the *latent* topics of the documents.

Pointwise Mutual Information (PMI) is very simple distributional measure that works well only in very large corpora. The similarity between two words w_1 and w_2 is given by the probability of seeing the two words together in a corpus divided by the probability of seeing them separately. This compensates for the chance of random co-occurrence when the words are frequent.

$$PMI(w_1, w_2) = log \frac{P(w_1, w_2)}{P(w_1)\ P(w_2)}$$

$$PMI(w_1, w_2) = log \frac{C(w1, w2)\ N}{C(w_1)\ C(w_2)}$$

The probabilities are simply the observed frequencies divided by N, the number of words in the corpus. We used the Web as a corpus, therefore we used the number of retrieved documents (hits returned by a search engine) to approximate the word co-occurrence counts, ignoring the fact that a word can be repeated in a document. Our experiments showed that using document counts instead of word counts leads to similar results.

A similarity measure that uses second-order co-occurrences (SOC-PMI) [10] works well even on a smaller corpus (BNC) because it looks at the words that co-occur with the two words. The method sort lists of important neighbor words of the two target words, using PMI, then it takes the shared neighbors and adds their PMI values, from the opposite list (normalizing by the number of neighbors).

2. Evaluation of Word Similarity Measures

Miller and Charles [22] asked several humans to judge the similarity of 30 noun pairs, a subset of the 65 noun pairs judged in a similar way by Rubenstein and Goodenough [26]. Here are some examples of pairs and similarity values, on a scale of 0 to 4 (averaged over the human judges):

[2] http://clg.wlv.ac.uk/demos/similarity/

[3] http://www.cs.ualberta.ca/~lindek/demos.htm

[4] http://lsa.colorado.edu/

Method Name	Miller and Charles 30 Noun Pairs	Rubenstein and Goodenough 65 Noun Pairs
Cosine (BNC)	0.406	0.472
SOC-PMI (BNC)	0.764	0.729
PMI (Web)	0.759	0.746
Leacock & Chodorow (WN)	0.821	0.852
Roget	0.878	0.818

TABLE 1. Correlations of similarity measures with human judges.

```
gem, jewel, 3.84
coast, shore, 3.70
asylum, madhouse, 3.61
magician, wizard, 3.50
shore,woodland,0.63
glass,magician,0.11
```

An automatic similarity method is considered good if it produces values that correlate well with the human values (correlation close to 1). Correlations for several measures are presented in table 1. Corpus-based values tend to have lower correlations than WordNet-based measures, because WordNet has a well-developed noun hierarchy. Among the WordNet-based measures we listed only the one with the highest correlation, the Leacock & Chodorow measure [11]. The Roget measure also has a very good correlation. Among the corpus-based measures, SOC-PMI and PMI are good.

The correlation with the human judges is a recommended evaluation step, but not sufficient because it can be done only on a small set of noun pairs. It can be used to filter out measures that are not promising.

The task-based evaluation section is the most indicative. The similarity measures can be evaluated in one or more tasks. The best measure is the one that achieves the highest performance in the evaluation measure appropriated for the task. It could be the case that different measures perform best for different tasks. Three tasks are presented in section 3.

A third type of evaluation measure consists in building an automatic thesaurus, by selecting a small number of close semantic neighbors for each word. Retrieval of semantic neighbors can be evaluated as in information retrieval systems [27]. The expected solution is an existing manually-built resource. A problem with this method is that resources tend to have different coverage.

3. Applications

3.1. **Solving TOEFL-style Synonym Questions.** A task commonly used in the evaluation of similarity measure is solving TOEFL-style questions. Two datasets

Method Name	Number of Correct Test Answers	Question/Answer Words Not Found	Percentage of Correct Answers
Roget	63	26	78.75%
SOC-PMI	61	4	76.25%
PMI-IR	59	0	73.75%
LSA	51.5	0	64.37%
Lin	32	42	40.00%

TABLE 2. Results on the 80 TOEFL Questions.

Method Name	Number of Correct Test Answers	Question/Answer Words Not Found	Percentage of Correct Answers
Roget	41	2	82%
SOC-PMI	34	0	68%
PMI-IR	33	0	66%
Lin	32	8	64%

TABLE 3. Results on the 50 ESL Questions.

are available: 80 synonym test questions from the Test of English as a Foreign Language (TOEFL) and 50 synonym test questions from a collection of English as a Second Language (ESL). An example of TOEFL question is:

```
The Smiths decided to go to Scotland for a short ..... They have already
booked return bus tickets.
      (a) travel
      (b) trip
      (c) voyage
      (d) move
```

The solution is one of the four choices that fits best into the context of the two sentences. The similarities between a choice word and each of the content words[5] in the sentences are added up, and the choice with the highest values is considered the solution. The results for the TOEFL questions results are presented in table 2 [10]. The results for the ESL questions are presented in table 3. The similarity measures from the tables are: Roget similarity [11], PMI-IR [29], SOC-PMI [10], LSA [13], and Lin [19]. The last one performs worse because many words were not available in the resource (a database of dependency relations). The best performance is achieved by the Roget measure.

3.2. **Detecting Speech Recognition Errors.** Another tasks is the detection of the words that do not fit into their context. For example, a spell-checker will not signal out words that are valid words but not the intended words. For example a

[5]We ignore function words such as prepositions, conjunctions, etc.

user could types *raw and column* when it was meant *row and column*. The task of real-word error correction [4] would detect that *raw* is a mistake, and suggest that *row* has higher similarity with the other words in the text than *raw*.

We applied this idea to the task of detecting speech recognition errors [9], as words that have low semantic similarity with their context. The data we used is 100 stories from the TDT corpus, which had manual transcripts. The automatic speech transcripts were produced with the BBN speech recognizer and had a word error rate of about 25%. Here is an example of automatic transcript and the corresponding manual transcript:

```
BBN transcript: time now for a geography was they were traveling down river
to a city that like many russian cities has had several names but this one
stanza is the scene of ethnic and national and world war two in which the
nazis were nine elated

Manual transcript: Time now for our geography quiz today. We're traveling
down the Volga river to a city that, like many Russian cities, has had
several names. But this one stands out as the scene of an epic battle in
world war two in which the Nazis were annihilated.

Detected outliers:  stanza, elated
```

Our algorithm detected two words as potential errors (semantic outliers). For each word w in the automatic transcript, the algorithm executed the following steps:

(1) Compute the neighborhood $N(w)$, i.e. the set of content words that occur close to w in the transcript (include w).
(2) Compute pair-wise semantic similarity scores $S(w_i, w_j)$ between all pairs of words $w_i \neq w_j$ in $N(w)$, using a semantic similarity measure.
(3) Compute the semantic coherence $SC(w_i)$ by adding the pair-wise semantic similarities $S(w_i, w_j)$ of w_i with all its neighbors $w_j \neq w_i$ in $N(w)$.
(4) Let SC_{avg} be the average of $SC(w_i)$ over all w_i in the neighborhood $N(w)$.
(5) Label w as a recognition errors if $SC(w) \leq K\ SC_{avg}$.

The neighborhood of a word could be the whole speech segment or part of it (a context window). The average coherence of the segment times a parameter K is used for comparison, as a threshold for signaling semantic outliers.

We varied the parameter K in order to detect more or fewer semantic outliers as potential speech recognition errors. Detecting too many brings the risk of signaling words that are not really speech recognition errors. We evaluated the performance in terms of the precision of the detected outliers and of their recall. The results in figure 1 show that using the PMI similarity measure (computed in the Waterloo Multitext corpus of Web data) leads to better results than using the

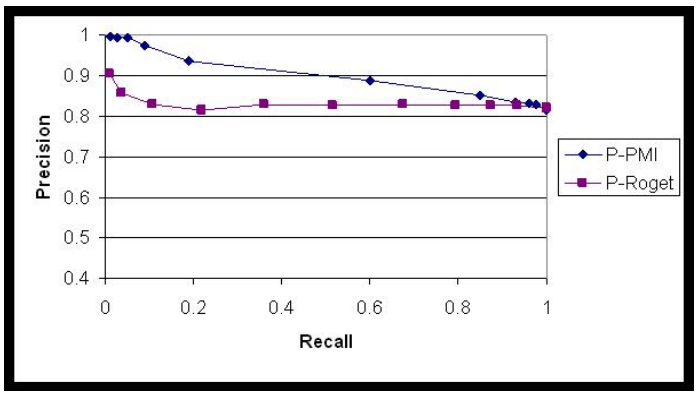


FIGURE 1. Results for detecting speech recognition errors.

Roget similarity measure. The Roget measure performed worse because some of the words were not found in the thesaurus.

3.3. **Synonym Choice in an Intelligent Thesaurus.** A third task that we describe concerns synonym choice in context, for writing aid tools. We developed an intelligent thesaurus [8], that allows a writer to select a word and to ask for synonym that would be alternative choices. There is a thesaurus is Microsoft Word that allows the writer to do this, but it does not order the choices by their suitability. Our thesaurus computes for each choice its similarity to the context, and orders the choices by these values. This helps the user to select the best choice.

In order to evaluate the method, we selected sentences and took out a word, creating a gap. Then we found synonyms for that word, and computed their similarity to the context. If the highest ranked synonym is exactly the word that we took out (the word that was in the original sentence), we consider that the recommendation of the intelligent thesaurus was correct.

Here are two examples of sentences and synonym sets. For the first one the original word was *error*, for the second one it was *job*.

```
Sentence: This could be improved by more detailed consideration of the
processes of ......... propagation inherent in digitizing procedures.
Solution set: mistake, blooper, blunder, boner, contretemps, error, faux pas,
goof, slip, solecism

Sentence:    The effort required has had an unhappy effect upon his prose,
on his ability to make the discriminations the complex ......... demands.
Solution set: job, task, chore
```

Test set	Baseline Most Freq. Syn.	Edmonds, 1997	Accuracy First Choice	Accuracy First Two Choices
Data set 1 Syns: WordNet (7 groups) Sentences: WSJ	44.8%	55%	66.0%	88.5%
Data set 2 Syns: CTRW (11 groups) Sentences: BNC	57.0%	–	76.5%	87.5%

TABLE 4. Results for the intelligent thesaurus.

We used the PMI measure with the Waterloo Multitext corpus and a context window of k content words before the gap and k words after the gap (k=2 was the best value, determined experimentally).

The results are presented in table 4. The first dataset used newspaper sentences (WSJ) and synonyms form WordNet. Our results were much better than a baseline of always choosing the most frequent synonym, and than a pervious method of Edmonds [5] that uses a lexical co-occurrence network. We improve over the baseline also on a second dataset, with sentences from the BNC and synonyms from a special dictionary of synonyms named *Choose the Right Word* (CTRW).

4. Text Similarity

The similarity of two texts can be computed in several ways, including the classic vector space model. Applications of text similarity knowledge include designing exercises for second language-learning, acquisition of domain-specific corpora, information retrieval, and text categorization.

Here we present a method for computing the similarity of two short texts, based on the similarities of their words. We used the SOC-PMI corpus-based similarity for two words. In addition, we used string similarity (longest common subsequence). The method selects a word from the first text and a word from the second text, which have the highest similarity. The similarity value is stored, and the two words are taken out. The method continues until there are no more words. At the end, the similarity scores are added and normalized.

For evaluation we used a data set of 30 sentence pairs for which similarity values computed by human judges were available [16]. In Figure 2 we present the correlation between the scores produced by our method and the average of the scores given by the human judges. Our results are better than the results of the method of Li et al. [16], based on a lexical co-occurrence network. The last two bars in the figure show how much the human judges varied from their mean.

The second dataset that we used for evaluation was the Microsoft Paraphrases corpus. It contains pairs of sentences that are marked as being paraphrases or not.

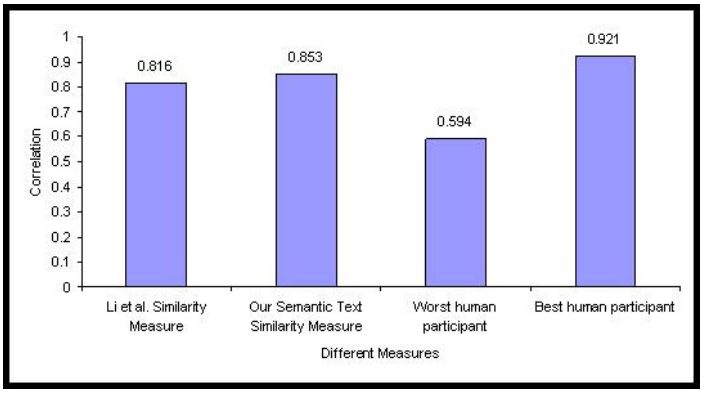


FIGURE 2. Correlation with human judges on the 30 sentence pairs.

Metric	Accuracy	Precision	Recall	F-measure
Random Baseline	51.3	68.3	50.0	57.8
Vector-based	65.4	71.6	79.5	75.3
Jiang & Conrath	69.3	72.2	87.1	79.0
Leacock & Chodorow	69.5	72.4	87.0	79.0
Lesk	69.3	72.4	86.6	78.9
Lin	69.3	71.6	88.7	79.2
Wu & Palmer	69.0	70.2	92.1	80.0
Resnik	69.0	69.0	96.4	80.4
Combined (Supervised)	71.5	72.3	92.5	81.2
Combined (Unsupervised)	70.3	69.6	97.7	81.3
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
STS	**72.6**	**74.7**	**89.1**	**81.3**

TABLE 5. Results on the MicroSoft Paraphrases corpus.

In this case we can evaluate if our method considers the two sentences as similar or not, we cannot evaluate the scores themselves.

Table 5 compares our results (the last line – Semantic Text similarity – STS) with the results obtained by Mihalcea et al. [21] on the same dataset. They used several WordNet-based measures, and combinations of these measures. We also compare to the PMI-IR and LSA corpus-based similarity measures. Our results are similar or slightly better than those of other methods.

5. Conclusion and Future Work

We presented an overview of the methods for computing word similarity. We discussed several ways to evaluate them. The main one is to evaluate them by how well they perform when solving specific tasks. We looked at three particular applications. We also discussed methods of computing the similarity of two short texts based on the similarity of their words.

There are several directions for future work. We plan to extend our second-order co-occurrences similarity measure to use a Web corpus, specifically the Google 5-gram corpus. This measure is promising because it worked well on the BNC

More investigation is needed in combining word similarity methods, in order to produce hybrid methods that use very large corpora. Such corpora are not annotated with WordNet senses. Automatic words sense disambiguation methods, though not powerful enough in general, could be sufficient for gathering statistics on word sense distribution in very large corpora.

We plan to develop cross-language similarity methods, for two words in different languages. If the two words are translations of each other, their similarity is maximal. If they are not translation the similarity could vary between zero and a value close to 1. For example, the similarity between the French word *pomme* and the English word *orange* can be computed by simply translating the French word into English (let's say the translations are *apple*, *potato*, and *head*), and take the maximum similarity between the translations and the second word. In this case the cross-language similarity is reduced to the similarity between the English words *apple* and *orange*. All is needed is a bilingual dictionary with a good coverage.

The cross-language similarity of two texts can be computed in the same way as the similarity of two texts in the same language, by using the similarity between the words (the cross-language word similarity measure that we sketched above). The cross-language similarity of two texts could be used in second language teaching to select similar texts, or in cross-language information retrieval.

References

[1] P.F. Brown, P.V. DeSouza, R.L. Mercer, T.J. Watson, V.J. Della Pietra, and J.C. Lai. Class-based n-gram models of natural language. Computational Linguistics, 18:467-479, 1992.

[2] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In Proceedings of IJCAI 2003.

[3] C. Buckley, J.A. Salton and A. Singhal. Automatic query expansion using Smart: TREC 3. In The third Text Retrieval Conference, Gaithersburg, MD, 1995.

[4] A. Budanitsky and G. Hirst. Evaluating WordNet-based measures of semantic distance. Computational Linguistics, 32(1), 2006.

[5] P. Edmonds. Choosing the word most typical in context using a lexical co-occurrence network. In Proceedings of ACL 1997.

[6] G. Grefenstette. Automatic thesaurus generation from raw text using knowledge-poor techniques. In Making Sense of Words, 9th Annual Conference of the UW Centre for the New OED and Text Research, 1993.

[7] G. Hirst and D. St-Onge. Lexical Chains as representations of context for the detection and correction of malapropisms. In WordNet An Electronic Database, 1998.
[8] D. Inkpen. Near-synonym choice in an Intelligent Thesaurus, HLT-NAACL 2007.
[9] D. Inkpen and A. Desilets. Semantic similarity for detecting recognition errors in automatic speech transcripts. In Proceedings of EMNLP 2005.
[10] A. Islam and D. Inkpen. Second order co-occurrence PMI for determining the semantic similarity of words. In Proceedings of LREC 2006.
[11] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. In Proceedings of RANLP 2003.
[12] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of COLING 1997.
[13] T.K. Landauer and S.T. Dumais. A Solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104(2), 1997.
[14] C. Leacock and M. Chodorow. Combining local context and WordNet sense similarity for word sense identification. In WordNet, An Electronic Lexical Database, 1998.
[15] Lesk, M.E. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the SIGDOC Conference , Toronto, 1986.
[16] Y. Li, D. McLean, Z. Bandar, J. O'Shea, K. and Crockett. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowledge and Data Eng. 18:8, 2006.
[17] H. Li and N. Abe. Word clustering and disambiguation based on co-occurrence data. In Proceedings of COLING-ACL, 1998, pp. 749-755.
[18] D. Lin. An information-theoretic definition of similarity. In Proceedings of ICML 1998.
[19] D. Lin. Automatic retrieval and clustering of similar words. In Proceedings of COLING-ACL, 1998, pp. 768-774.
[20] C.Y. Lin and E.H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of HLT-NAACL, 2003.
[21] R. Mihalcea, C. Corley, C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of AAAI 2006.
[22] G.A. Miller and W.G. Charles. Contextual correlates of semantic similarity. Language and Cognitive Processes, 6(1): 1-28, 1991.
[23] P. Pantel and D. Lin. Discovering word senses from text. In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2002, pp. 613-619.
[24] S. Patwardhan. Incorporating dictionary and corpus information into a vector measure of semantic relatedness. MSc Thesis, University of Minnesota, 2003.
[25] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its applications to problems of ambiguity in natural language. JAIR 11, 1999.
[26] H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. Communications of the ACM, 8(10): 627-633, 1995.
[27] J. Weeds, D. Weir and D. McCarthy. Characterising measures of lexical distributional similarity. In Proceedings of COLING 2004.
[28] Z. Wu and M. Palmer. Verb semantics and lexical selection. In Proceedings of ACL 1994.
[29] P.D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of ECML 2001.
[30] J. Xu and B. Croft. Improving the effectiveness of information retrieval. ACM Transactions on Information Systems, 18(1):79-112, 2000.

University of Ottawa, School of Information Technology and Engineering
E-mail address: diana@site.uottawa.ca

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

AN ALGORITHM FOR DETERMINATION OF NASH EQUILIBRIA IN THE INFORMATIONAL EXTENDED TWO-MATRIX GAMES

NOVAC LUDMILA

Abstract. In this article the informational extended games ${}_1\Gamma$ and ${}_2\Gamma$ are defined. For these informational extended two-matrix games we present two modes for construction of the extended matrices and an algorithm for determination of Nash equilibria. For this algorithm we make some modifications and present an algorithm for determination Nash equilibria in the informational extended two-matrix games in the case, in which the dimensions of the matrices are too big. Using this algorithm we can also determine the number of Nash equilibria in informational extended game, without using of the extended matrices.

Last years the informational aspect represents a real fillip for the elaboration of the new study methods for the non-cooperative game theory. The informational aspect in the game theory is manifested by: the devise of possession information about strategy's choice, the payoff functions, the order of moves, and optimal principles of players; the using methods of possessed information in the strategy's choice by players. The inclusion of information as an important element of game have imposed a new structure to the game theory: the games in complete information (the games in extended form), the games in not complete information and the games in imperfect information (the Bayes games). The player's possession of supplementary information about unfolding of the game can influence appreciably the player's payoffs.

An important element for the players represents the possession of information about the behaviour of his opponents. Thus for the same sets of strategies and the same payoff functions it is possible to obtain different results, if the players have supplementary information. So the information for the players about the strategy's choice by the others players have a significant role for the unfolding of the game.

Let us consider the two-matrix game in the normal form $\Gamma = \langle N, X_1, X_2, A, B\rangle$, where $A = \{a_{ij}\}$, $B = \{b_{ij}\}$, $i = \overline{1,m}$, $j = \overline{1,n}$ (A and B are the payoff matrices for the first and the second player respectively. Each player can choose one of his strategies and his purpose is to maximize his payoff. The player can choose his strategy independently of his opponent and the player does not know the chosen strategy of his opponent.

According to [1] we will define the Nash equilibrium.

Definition 1. *The pair* (i^*, j^*), $i^* \in X_1, j^* \in X_2$ *is called Nash equilibrium (NE) for the game* Γ, *if the next relations hold*

$$\begin{cases} a_{i^*j^*} \geqslant a_{ij^*}, \forall i \in X_1, \\ b_{i^*j^*} \geqslant a_{i^*j}, \forall j \in X_2. \end{cases}$$

Notation: $(i^*, j^*) \in NE(\Gamma)$.

There are two-matrix games for which the set of the Nash equilibria is empty: $NE(\Gamma) = \emptyset$ (solutions do not exist in pure strategies).

For every two-matrix game we can construct some informational extended games. If one of the players knows the strategy chosen by the other, we consider that it is one form of the informational extended two-matrix game for the initial game. Even if the initial two-matrix game has no solutions in pure strategies, for the informational extended games at least one solution in pure strategies always exists (Nash equilibria). Proof of this assertion see in [2], [3]. In the case of informational extended games the player which knows the chosen strategy of his opponent has one advantage and he will obtain one of his greater payoff.

According to [1], let us define two forms of informational extended games ${}_1\Gamma$ and ${}_2\Gamma$. We consider that for the game ${}_1\Gamma$ the first player knows the chosen strategy of the second player, and for the game ${}_2\Gamma$ the second player knows the chosen strategy of the first player.

If one of the players knows the chosen strategy of the other, then the set of the strategies for this player can be represented by a set of mappings defined on the set of strategies of his opponent.

Definition 2. *(The game* ${}_1\Gamma$ according to [1]*) The informational extended two-matrix game* ${}_1\Gamma$ *can be defined in the normal form by:* ${}_1\Gamma = \left\langle N, \overline{X_1}, X_2, \overline{A}, \overline{B} \right\rangle$, *where* $N = \{1, 2\}$, $\overline{X_1} = \{\varphi_1 : X_2 \longrightarrow X_1\}$, $\overline{A} = \{\overline{a}_{ij}\}$, $\overline{B} = \left\{\overline{b}_{ij}\right\}$, $i = \overline{1, m^n}$, $j = \overline{1, n}$.

For the game ${}_1\Gamma$ we have $\overline{X_1} = \{1, 2, \ldots, m^n\}$, $X_2 = \{1, 2, \ldots, n\}$, $\left|\overline{X_1}\right| = m^n$, and the matrices $\overline{A}$ and $\overline{B}$ have dimension $[m^n \times n]$ and are formed from elements of initial matrices A and B respectively.

The matrices $\overline{A}$ and $\overline{B}$ will be constructed in the next mode:

Let us denote by $A_{i\cdot} = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$, $B_{i\cdot} = \{b_{i1}, b_{i2}, \ldots, b_{in}\}$, $i = \overline{1, m}$ the rows i in the matrices A and B, respectively).

Choosing one element from each of these rows $A_{1\cdot}, A_{2\cdot}, \ldots, A_{m\cdot}$, we will build one column in the matrix $\overline{A}$. The columns from the matrix $\overline{B}$ are built in the same mode, choosing one element from each of the rows $B_{1\cdot}, B_{2\cdot}, \ldots, B_{m\cdot}$.

Thus, the matrices $\overline{A}$ and $\overline{B}$ have the dimension $[m^n \times n]$.

Definition 3. *(The game* ${}_2\Gamma$ according to [1]*) The informational extended two-matrix game* ${}_2\Gamma$ *can be defined in the normal form by:* ${}_2\Gamma = \left\langle N, X_1, \overline{X_2}, \widetilde{A}, \widetilde{B} \right\rangle$,

where $\overline{X_2} = \{\varphi_2 : X_1 \longrightarrow X_2\}$, $\left|\overline{X_2}\right| = n^m$, $\widetilde{A} = \{\widetilde{a}_{ij}\}$, $\widetilde{B} = \left\{\widetilde{b}_{ij}\right\}$, $i = \overline{1,m}$, $j = \overline{1,n^m}$.

For the game ${}_2\Gamma$ we have $X_1 = \{1,2,\ldots,m\}, \overline{X_2} = \{1,2,\ldots,n^m\}$ and the matrices $\widetilde{A}$ and $\widetilde{B}$ have dimension $[m \times n^m]$ and are formed from elements of initial matrices A and B respectively.

The extended matrices $\widetilde{A}$ *and* $\widetilde{B}$ will be built in analogical mode as in the case of the game ${}_2\Gamma$.

Let us denote by $A_{\cdot j} = \{a_{1j}, a_{2j}, \ldots, a_{mj}\}$, $B_{\cdot j} = \{b_{1j}, b_{2j}, \ldots, b_{mj}\}$, $j = \overline{1,n}$ the columns j in the initial matrices A and B, respectively). Each of rows in the matrices $\widetilde{A}$ (or in the matrix $\widetilde{B}$, respectively) will be built choosing one element from each of the columns $A_{\cdot j}$ (or from the columns $B_{\cdot j}$, respectively).

The next theorem represents the condition of the Nash equilibria existence for the informational extended two-matrix games ${}_1\Gamma$ and ${}_2\Gamma$.

Theorem 1. *For every two-matrix game* Γ *we have the following*

$$NE\left({}_1\Gamma\right) \neq \emptyset,\ NE\left({}_2\Gamma\right) \neq \emptyset;\ \textit{and}\ NE\left(\Gamma\right) \subset\ NE\left({}_1\Gamma\right),\ NE\left(\Gamma\right) \subset\ NE\left({}_2\Gamma\right).$$

For proof see [2], [3].

For the informational extended games ${}_1\Gamma$ and ${}_2\Gamma$ we can proof the following statements.

Assertion 1. *If* $\exists i^* \in X_1$, $\exists j^* \in X_2$ *for which* $a_{i^*j^*} = \max\limits_i \max\limits_j a_{ij}$, $b_{i^*j^*} = \min\limits_i \min\limits_j b_{ij}$ *and* $\forall i \in X_1$, $\forall j \in X_2 : (i,j) \neq (i^*,j^*)$ *so that* $a_{ij} < a_{i^*j^*}$, $b_{ij} > b_{i^*j^*}$; *then:*

1) in the game ${}_2\Gamma$ *all columns* k *(from* $\widetilde{A}$ *which contain the element* $a_{i^*j^*}$, *and from* $\widetilde{B}$ *which contain the element* $b_{i^*j^*}$*) do not contain NE equilibria*;

2) in the game ${}_1\Gamma$ *the column* j^* *(in the matrices* $\overline{A}$ *and* $\overline{B}$*) do not contains NE equilibria.*

Assertion 2. *If* $\exists i^* \in X_1$, $\exists j^* \in X_2$ *so that* $a_{i^*j^*} = \min\limits_i \min\limits_j a_{ij}$ *and* $b_{i^*j^*} = \max\limits_i \max\limits_j b_{ij}$, *and* $\forall i \in X_1$, $\forall j \in X_2 : (i,j) \neq (i^*,j^*)$ *so that* $a_{ij} > a_{i^*j^*}$, $b_{ij} < b_{i^*j^*}$; *then:*

1) in the game ${}_2\Gamma$ *the row* i^* *(the* $\widetilde{A}_{i^*\cdot}$ *and the* $\widetilde{B}_{i^*\cdot}$*) does not contain NE equilibria*;

2) in the game ${}_1\Gamma$ *all rows* k *(the* $\overline{A}_{k\cdot}$, *and the* $\overline{B}_{k\cdot}$ *which contain the elements* $a_{i^*j^*}$ *and* $b_{i^*j^*}$, *respectively) do not contain NE equilibria.*

From the assertions 1 and 2 the next two statements result.

Assertion 3. *Consider that* $\exists i^* \in X_1$, $\exists j^* \in X_2$ *so that* $a_{i^*j^*} = \max\limits_i \max\limits_j a_{ij}$ *and* $b_{i^*j^*} = \min\limits_i \min\limits_j b_{ij}$.

1) If $\forall i \in X_1 \setminus \{i^*\}$, $\forall j \in X_2 : a_{ij} < a_{i^*j^*}$, *and* $\forall j \in X_2 \setminus \{j^*\} : b_{i^*j} > b_{i^*j^*}$, *then in the game* ${}_2\Gamma$ *each of columns* k *(*$\widetilde{A}_{\cdot k}$, $\widetilde{B}_{\cdot k}$ *which contains the elements* $a_{i^*j^*}$ *and* $b_{i^*j^*}$, *respectively) does not contain NE equilibria.*

2) If $\forall i \in X_1, \forall j \in X_2 \setminus \{j^*\} : b_{ij} > b_{i^*j^*}$ *and* $\forall i \in X_1 \setminus \{i^*\} : a_{ij^*} < a_{i^*j^*}$, *then in the game* ${}_1\Gamma$ *the column* j^* *(*$\overline{A}_{\cdot j^*}$ *and* $\overline{B}_{\cdot j^*}$*) does not contain NE equilibria.*

Assertion 4. *Consider that* $\exists i^* \in X_1$, $\exists j^* \in X_2$ *so that* $a_{i^*j^*} = \min\limits_i \min\limits_j a_{ij}$ *and* $b_{i^*j^*} = \max\limits_i \max\limits_j b_{ij}$.

1) If $\forall i \in X_1 \setminus \{i^*\}$, $\forall j \in X_2 : a_{ij} > a_{i^*j^*}$, *and* $\forall j \in X_2 \setminus \{j^*\} : b_{i^*j} < b_{i^*j^*}$, *then in the game* ${}_2\Gamma$ *the row* i^* *(*$\widetilde{A}_{i^*\cdot}$, $\widetilde{B}_{i^*\cdot}$*) does not contain NE equilibria.*

2) If $\forall i \in X_1$, $\forall j \in X_2 \setminus \{j^*\} : b_{ij} < b_{i^*j^*}$ *and* $\forall i \in X_1 \setminus \{i^*\} : a_{ij^*} > a_{i^*j^*}$, *then in the game* ${}_1\Gamma$ *each of rows* k *(*$\overline{A}_{k\cdot}$, $\overline{B}_{k\cdot}$ *which contains the elements* $a_{i^*j^*}$ *and* $b_{i^*j^*}$, *respectively) does not contain NE equilibria.*

Example 1. (For Assertions 2 and 4).

$A = \begin{pmatrix} 0 & 3 & 1 \\ 5 & 2 & 4 \end{pmatrix}$, $B = \begin{pmatrix} 7 & 3 & 6 \\ 1 & 5 & 0 \end{pmatrix}$.

For this game $NE(\Gamma) = \emptyset$.

For the game ${}_2\Gamma$ there are two Nash equilibria $(2,2), (2,8) \in NE\,({}_2\Gamma)$.

For the game ${}_1\Gamma$ there is only one Nash equilibrium $(6,2) \in NE\,({}_1\Gamma)$.

In this game, for $i = 1$, $j = 1 : \min\limits_i \min\limits_j a_{ij} = 0$, $\max\limits_i \max\limits_j b_{ij} = 7$. According to Assertion 2 and 4, it follows that: for the game ${}_2\Gamma$ the first row does not contain Nash equilibria and for the game ${}_1\Gamma$ the 1^{st}, 2^d, 3^d, 4^{th} rows do not contain Nash equilibria.

$\widetilde{A} = \begin{pmatrix} 0 & 0 & 0 & 3 & 3 & 3 & 1 & 1 & 1 \\ 5 & \underline{\mathbf{2}} & 4 & 5 & 2 & 4 & 5 & \underline{\mathbf{2}} & 4 \end{pmatrix}$, $\widetilde{B} = \begin{pmatrix} 7 & 7 & 7 & 3 & 3 & 3 & 6 & 6 & 6 \\ 1 & \underline{\mathbf{5}} & 0 & 1 & 5 & 0 & 1 & \underline{\mathbf{5}} & 0 \end{pmatrix}$.

$\overline{A} = \begin{pmatrix} 0 & 3 & 1 \\ 0 & 3 & 4 \\ 0 & 2 & 1 \\ 0 & 2 & 4 \\ 5 & 3 & 1 \\ 5 & \underline{\mathbf{3}} & 4 \\ 5 & 2 & 1 \\ 5 & 2 & 4 \end{pmatrix}$, $\overline{B} = \begin{pmatrix} 7 & 3 & 6 \\ 7 & 3 & 0 \\ 7 & 5 & 6 \\ 7 & 5 & 0 \\ 1 & 3 & 6 \\ 1 & \underline{\mathbf{3}} & 0 \\ 1 & 5 & 6 \\ 1 & 5 & 0 \end{pmatrix}$.

Example 2. (For Assertions 1 and 3).

$A = \begin{pmatrix} 7 & 3 & 6 \\ 1 & 5 & 0 \end{pmatrix}$; $B = \begin{pmatrix} 0 & 3 & 1 \\ 4 & 2 & 5 \end{pmatrix}$.

For this game $NE(\Gamma) = \emptyset$, and for the informational extended games there are some solutions $(1,4), (1,6) \in NE\,({}_2\Gamma)$, $(3,2) \in NE\,({}_1\Gamma)$.

In this game, for $i = 1$, $j = 1 : \max\limits_i \max\limits_j a_{ij} = 7$, $\min\limits_i \min\limits_j b_{ij} = 0$. According to Assertions 1 and 3, it follows that: for the game ${}_2\Gamma$ the 1^{st}, 2^d, 3^d columns do not contain Nash equilibria and for the game ${}_1\Gamma$ the first column does not contain Nash equilibria.

$$\widetilde{A} = \begin{pmatrix} 7 & 7 & 7 & \underline{\mathbf{3}} & 3 & \underline{\mathbf{3}} & 6 & 6 & 6 \\ 1 & 5 & 0 & 1 & 5 & 0 & 1 & 5 & 0 \end{pmatrix} \quad \overline{A} = \begin{pmatrix} 7 & 3 & 6 \\ 7 & 3 & 0 \\ 7 & \underline{\mathbf{5}} & 6 \\ 7 & 5 & 0 \\ 1 & 3 & 6 \\ 1 & 3 & 0 \\ 1 & 5 & 6 \\ 1 & 5 & 0 \end{pmatrix}, \ \overline{B} = \begin{pmatrix} 0 & 3 & 1 \\ 0 & 3 & 5 \\ 0 & \underline{\mathbf{2}} & 1 \\ 0 & 2 & 5 \\ 4 & 3 & 1 \\ 4 & 3 & 5 \\ 4 & 2 & 1 \\ 4 & 2 & 5 \end{pmatrix}.$$

$$\widetilde{B} = \begin{pmatrix} 0 & 0 & 0 & \underline{\mathbf{3}} & 3 & \underline{\mathbf{3}} & 1 & 1 & 1 \\ 4 & 2 & 5 & 4 & 2 & 5 & 4 & 2 & 5 \end{pmatrix}$$

Example 3. (For Assertions 2 and 4).

$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 6 \\ 3 & 1 \\ 1 & 4 \end{pmatrix}$, $NE(\Gamma) = \emptyset$.

$$\widetilde{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & \underline{\mathbf{0}} & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & \underline{\mathbf{0}} & 2 & 2 \\ 4 & 0 & 4 & 0 & 4 & \underline{\mathbf{0}} & 4 & 0 \end{pmatrix}, \ \widetilde{B} = \begin{pmatrix} 2 & 2 & 2 & 2 & 6 & \underline{\mathbf{6}} & 6 & 6 \\ 3 & 3 & 1 & 1 & 3 & \underline{\mathbf{3}} & 1 & 1 \\ 1 & 4 & 1 & 4 & 1 & \underline{\mathbf{4}} & 1 & 4 \end{pmatrix}.$$

In this game, for the pairs $(i^*, j^*) : (1,2)\,, (2,1)\,, (3,2)$ we have $\min\limits_{i} \min\limits_{j} a_{ij} = 0$, and for each row $\max\limits_{j} b_{ij} = b_{i^*j^*}$, but because each of rows from the matrix A contains the minimum element $a_{12} = 0$, for the 6^{th} column the conditions from assertions 2 and 4 do not hold , and $(1,6)\,, (2,6), (3,6) \in NE\,({}_2\Gamma)\,.$ □

For the generation of the extended matrices $\overline{A}$ and $\overline{B}$ (or the $\widetilde{A}$ and the $\widetilde{B}$, respectively) we can use the next methods.

The first method is based on representation of decimal numbers in the base which represent the number of rows or the number of columns in the initial matrices.

For the game ${}_1\Gamma$ we need to represent the numbers $0, 1, \ldots, (m^n - 1)$ in the base m with n components: $N_m = (C_0 C_1 \ldots C_{n-1})_m$, where $C_j \in \{0, 1, \ldots, m-1\}$, $j = \overline{0, n-1}$, that is $\left(C_0 m^0 + C_1 m^1 + \ldots + C_{n-1} m^{n-1}\right) = N_{10}$. Each of these numbers N_m represented in the base m will correspond to one column in the extended matrix.

Then for elements from column j it must replace:

$0 \to a_{1j}, 1 \to a_{2j}, \ldots, i \to a_{(i+1)j}, \ldots, (m-1) \to a_{mj}$ (similarly for the matrix B).

For the game ${}_2\Gamma$ it must represent the numbers $0, 1, \ldots, (n^m - 1)$ in the base n with m components: $N_n = (C_0 C_1 \ldots C_{m-1})_n$, where $C_i \in \{0, 1, \ldots, n-1\}$, $i = \overline{0, m-1}$, that is $\left(C_0 n^0 + C_1 n^1 + \ldots + C_{m-1} n^{m-1}\right) = N_{10}$. Each of these numbers N_n represented in the base n will correspond to one row into the extended matrix.

Then for the elements from the row i it must replace:

$0 \to a_{i1}, 1 \to a_{i2}, \ldots, j \to a_{i(j+1)}, \ldots, (n-1) \to a_{in}$ (similarly for the matrix B).

The second method consists in assigning two numbers to each of the elements from the initial matrices. One of these numbers represents the number of blocks (series) formed by this element, and the second number represents the length of the block (that is, the number of repetitions of this element in the block).

Denote by $nrbl$ the number of blocks for some element a_{ij} (b_{ij}) and by L the length of each of blocks (the number of repetitions of this element in the block).

Thus for the game ${}_2\Gamma$ assign to each element from the row i: (n^{i-1}) blocks (series) each of them with length (n^{m-i}).

So for all elements a_{ij}, b_{ij}, $i = \overline{1,m}, j = \overline{1,n}$ we can determine the indices of columns k of this element in the extended matrix. Thus for the element from the row i and from the column j and for all $nrbl = \overline{1,n^{i-1}}$, $L = \overline{1,n^{m-i}}$, we calculate the number k by:

$$k = n \cdot n^{m-i} \cdot (nrbl - 1) + (j-1) \cdot n^{m-i} + L. \tag{1}$$

In such mode we can construct the extended matrices $\widetilde{A}$ and $\widetilde{B}$: $\widetilde{A}[i,k] = A[i,j]$, $\widetilde{B}[i,k] = B[i,j]$.

Similarly, for the game ${}_1\Gamma$ we assign to each element from the column j: (m^{j-1}) blocks (series) each of them with length (m^{n-j}).

Thus for all elements $\forall i = \overline{1,m}, j = \overline{1,n}$, we determine the indices of the rows k of this element in the extended matrix.

In such mode for the element from the row i and from the column j and for all $nrbl = \overline{1,m^{j-1}}$, $L = \overline{1,m^{n-j}}$ we calculate the number k by:

$$k = m \cdot m^{n-j} \cdot (nrbl - 1) + (i-1) \cdot m^{n-j} + L. \tag{2}$$

In such mode, we can construct the extended matrices $\overline{A}$ and $\overline{B}$ (for each determined k): $\overline{A}[k,j] = A[i,j]$, $\overline{B}[k,j] = B[i,j]$.

Remark. These two different methods may be used independently. Using it we can construct the extended matrices entirely or partly. If the initial matrices are very big, we can use these methods for partial construction of the extended matrices. Thus the first method may be used when we need to construct only one row (for the informational extended game ${}_1\Gamma$), or only one column (for the game ${}_2\Gamma$), and the second method may be used when we need to determine the position of some element in the extended matrix, i.e. the index of the row (in the game ${}_1\Gamma$) or the index of the column (in the game ${}_2\Gamma$, respectively).

Example 4. (The generation of the extended matrices).

$A = \begin{pmatrix} 0 & 3 & 1 \\ 5 & 2 & 4 \end{pmatrix}$, $B = \begin{pmatrix} 7 & 3 & 6 \\ 1 & 5 & 0 \end{pmatrix}$ $m = 2$, $n = 3$.

For the **first** method:

For the game ${}_1\Gamma$ the matrices are of dimension $\left[2^3 \times 3\right]$. We construct the 5^{th} row from the extended matrix $\overline{A}$:

$4_{10} = (100)_2$, next we do the substitution with corresponding elements and we obtain the 5^{th} row with elements (5,3,1).

In the same mode we can construct the 8^{th} row: $7_{10} = (111)_2$ and we obtain the row (1,5,0) from the extended matrix $\overline{B}$.

For the game ${}_2\Gamma$ the matrices are of dimension $\left[2 \times 3^2\right]$. We construct the 6^{th} column:

$5_{10} = (12)_3$, next we do the substitution with corresponding elements and we obtain the 6^{th} column: (3,4) from the extended matrix $\widetilde{A}$ and the 6^{th} column (3,0) from the matrix $\widetilde{B}$.

In the same mode we can construct the 9^{th} column: $8_{10} = (22)_3$ and we obtain the columns (1,4) and (6,0) from the extended matrices ($\widetilde{A}$ and $\widetilde{B}$, respectively).

For the **second** method:

For the same game we determine the positions in the extended matrices for the elements $a_{21} = 5$ and $b_{21} = 1$.

For the game ${}_1\Gamma$ in the first column will contain one series (2^0 blocks) which will have 2^2 elements; the indices of rows are $k = 5, 6, 7, 8$.

For the game ${}_2\Gamma$ in the second row will contain (3^1) series (blocks) and each of them will have one element (i. e. 3^0 elements); the indices of columns are $k = 1, 4, 7$. $\square$

Using these methods we can construct an algorithm for determination of the NE equilibrium. This algorithm does not need the integral construction of the extended matrices, and need only the partial construction of them.

Thus in the case when the dimension of the initial matrices A and B are very big we avoid using a big volume of memory, since the extended matrices will have a bigger dimensions ($[m \times n^m]$ and $[m^n \times n]$, respectively).

The following algorithm can be used for determination of Nash equilibria in the informational extended two-matrix games ${}_1\Gamma$ and ${}_2\Gamma$.

Algorithm.

Consider the extended game ${}_2\Gamma$.

Using the first method we represent the numbers from 0 to $(n^m - 1)$ in the base n. Each of these representations will correspond to one column in the extended matrix $\widetilde{A}$. For each of these representations it must make the substitutions with the corresponding elements from the initial matrix A.

For each column $j_0 = \overline{1, n^m}$, obtained in such mode, from the extended matrix $\widetilde{A}$ we will do the next operations.

1. We determine the maximum element from this column of the extended matrix $\widetilde{A}$, and the corresponding element with the same indices from the matrix $\widetilde{B}$; let them $\widetilde{a}_{i_0 j_0}$ and $\widetilde{b}_{i_0 j_0}$.

2. We determine the maximum element from the row i_0 in the initial matrix B: let it be $b_{i_0 j^*}$.

3. If $\widetilde{b}_{i_0 j_0} = b_{i_0 j^*}$, .then (i_0, j_0) is NE equilibrium for the extended game ${}_2\Gamma$: $(i_0, j_0) \in NE\,({}_2\Gamma)$, and the elements $\widetilde{a}_{i_0 j_0}$ and $\widetilde{b}_{i_0 j_0}$ will be the payoff's values for the first and for the second player respectively.

For the informational extended game ${}_1\Gamma$ we can construct the algorithm in the same mode.

Consider now the extended game ${}_1\Gamma$.

Using the first method we represent the numbers from 0 to (m^n-1) in the base m. For each of these representations it must do the substitutions with the corresponding elements from the initial matrix B. Each of these representations will correspond to one row in the extended matrix $\overline{B}$.

For each row i_0 ($i_0=\overline{1,m^n}$) from the matrix $\overline{B}$ (thus obtained) we will do the next operations.

1. We determine the maximum element from this row of the extended matrix $\overline{B}$, and the corresponding element with the same indices from the matrix $\overline{A}$; let them be $\overline{b}_{i_0j_0}$ and $\overline{a}_{i_0j_0}$.

2. We determine the maximum element from the column j_0 in the initial matrix A : let's consider this element $a_{i^*j_0}$.

3. If $\overline{a}_{i_0j_0}=a_{i^*j_0}$, then (i_0,j_0) is NE equilibrium for the extended game ${}_1\Gamma$: $(i_0,j_0)\in NE\,({}_1\Gamma)$, and the elements $\overline{a}_{i_0j_0}$ and $\overline{b}_{i_0j_0}$ will be the payoff's values for the first and for the second player respectively.

Example 5.

$$A=\begin{pmatrix}2&5\\ \underline{\mathbf{4}}&1\\ 3&7\end{pmatrix},\ B=\begin{pmatrix}5&9\\ \underline{\mathbf{2}}&1\\ 6&4\end{pmatrix}.$$

This game has only one Nash equilibrium.

We can determine the Nash equilibria without using the extended matrices.

For the game ${}_2\Gamma$ we need to represent the numbers from 0 to $8=2^3$ in the base 2.

For the first column: 0_{10} =(0,0,0)$_2$ we do the substitution with corresponding elements (2,4,3), $\max\{2,4,3\}=4=a_{21}$, and the corresponding element b_{21} is the maximum element from the second row from the matrix B, thus follows that: $(2,1)\in NE({}_2\Gamma)$;

- for the second column : 1_{10} =(0,0,1)$_2$ the corresponding elements are (2,4,7), for which $\max\{2,4,7\}=7=a_{32}$, but the corresponding element $b_{32}\neq\max\{6,4\}$ from the third row of the matrix B, so $(3,2)\notin NE({}_2\Gamma)$;

- for the third column 2_{10}=(0,1,0)$_2$ for which $\max\{2,1,3\}=3=a_{31}$ we have $b_{31}=\max\{6,4\}$, thus $(3,3)\in NE({}_2\Gamma)$;

- for the 5^{th} column 4_{10}=(1,0,0)$_2$ we have $\max\{5,4,3\}=5=a_{12}$ and $b_{12}=\max\{5,9\}$, so it follows that $(1,5)\in NE({}_2\Gamma)$;

- for the 7^{th} column 6_{10} =(1,1,0)$_2$ we have $\max\{5,1,3\}=5=a_{12}$ and $b_{12}=\max\{5,9\}$, so $(1,7)\in NE({}_2\Gamma)$.

If we will build the extended matrices, we will see that for the informational extended game ${}_2\Gamma$ there are only four Nash equilibria.

$$\begin{pmatrix} 2 & 2 & 2 & 2 & \underline{\mathbf{5}} & 5 & \underline{\mathbf{5}} & 5 \\ \underline{\mathbf{4}} & 4 & 1 & 1 & 4 & 4 & 1 & 1 \\ 3 & 7 & \underline{\mathbf{3}} & 7 & 3 & 7 & 3 & 7 \end{pmatrix}, \qquad \begin{pmatrix} 5 & 5 & 5 & 5 & \underline{\mathbf{9}} & 9 & \underline{\mathbf{9}} & 9 \\ \underline{\mathbf{2}} & 2 & 1 & 1 & 2 & 2 & 1 & 1 \\ 6 & 4 & \underline{\mathbf{6}} & 4 & 6 & 4 & 6 & 4 \end{pmatrix}.$$

If we need to determine the indices of the columns in the extended matrices in the game ${}_2\Gamma$ for the elements a_{21}, b_{21}, and we know that $(2,1) \in NE(\Gamma)$, we can use relation (1) from the second method. So in this case indices of columns are $k = 1, 2, 5, 6$, but only one of these columns contains NE equilibrium $(2,1) \in NE({}_2\Gamma)$. $\square$

Remark. In the case when the numbers n^m and m^n are very big this algorithm for determination of NE equilibria for the informational extended games and the generation methods of the extended matrices are more complex. But all these operations can be executed operating with the corresponding numbers represented in the base m or n respectively to the informational extension (${}_1\Gamma$ or ${}_2\Gamma$ respectively).

The operating with numbers represented in the base n.

Consider the informational extended game ${}_2\Gamma$.

For the game ${}_2\Gamma$ the extended matrices will have dimensions $[m \times n^m]$ (by definition).

According to the second method, to each element from the row i two numbers correspond : $nrbl = n^{i-1}$ of blocks, each of them have the length $L = n^{m-i}$.

The relation (1) used in the second method for the game ${}_2\Gamma$ can be written in the next form:

$$k = n^{m-i} \cdot (n \cdot nrbl - n + (j-1)) + L. \tag{3}$$

We will represent all numbers from the relation (3) in the base n with m components:

$n = \left(00\ldots0\underset{2}{1}0\right)_n$;

$n^{i-1} = N_n = \left(0\ldots0\underset{i}{1}0\ldots0\underset{1}{}\right)_n$, $i = \overline{1,m}$;

$n^{m-i} = N_n = \left(0\ldots0\underset{m-i+1}{1}0\ldots0\underset{1}{}\right)_n$, $i = \overline{1,m}$;

the number of blocks is determined by: $nrbl = \overline{1, n^{i-1}}$, so

$nrbl = (00\ldots01)_n, \ldots, \left(0\ldots0\underset{i}{1}0\ldots0\underset{1}{}\right)_n$;

the length of blocks is determined by: $L = \overline{1, n^{m-i}}$, thus

$L = (00\ldots01)_n, \ldots, \left(0\ldots0\underset{m-i+1}{1}0\ldots0\underset{1}{}\right)_n$.

Using the relation (3) all operations can be done, operating with numbers represented in the base n.

Thus, using in the relation (3) the numbers represented in the base n, we determine k.

All arithmetic operations (*,+,-) will be executed in the base n.

Remark. The operation "*" in the base n for one number with other number in the form $\left(0\dots0\underset{i+1}{1}0\dots0\underset{1}{}\right)_n = n^i$ is equivalent to moving to the left with i positions of the components from the first number (so add i zeroes to the right).

Remark. The operations (+,-) for two numbers in the base n are done according to the well-known rules characteristic for the base 10.

Example 6.

Consider that the game Γ have matrices of dimension $[6\times 6]$, i. e. $m=6, n=6$, and we need to determine the index of the column k for the elements a_{25} and b_{25} in the extended matrices for the game ${}_2\Gamma$ (i. e. $i=2, j=5$), $m-i+1=5$; it's known that for the number of blocks (series) holds next ($1\leqslant nrbl\leqslant n^{i-1}=n$), so we have $nrbl=(0...01)_6,...,(0...010)_6$ in the base 6, and $n^{m-i}=(010000)_6$. Consider that $nrbl=000005$ and $L=(015355)_6$. Using the relation (3), all operations can be done operating with numbers represented in the base 6 :

$$\begin{array}{rl}
000005 & = nrbl\\
*\underline{000010} & = n\\
000050 & \\
+\underline{000004} & = j-1\\
000054 & \\
-\underline{000010} & = n\\
000044 & \\
*\underline{010000} & = n^{m-i}\\
440000 & \\
+\underline{015355} & = L\\
455355 & = k
\end{array}$$

Thus, we just have obtained one of the indices (represented in the base 6: $k=455355$) of the columns for the elements a_{25}, b_{25} in the extended matrices for the game ${}_2\Gamma$.

Remark. In this algorithm we can do operations in other order for determination Nash equilibria in the informational extended games ${}_1\Gamma$, ${}_2\Gamma$. Using this modified algorithm, we can determine also the number of Nash equilibria in the games ${}_1\Gamma$, ${}_2\Gamma$, without using of the extended matrices. Thus for the game ${}_1\Gamma$, (${}_2\Gamma$) firstly we determine the maximum payoff for the first (second) player and the corresponding strategy for this maximum element; then we determine the corresponding combinations for that we obtain the maximum payoff and the corresponding strategy for the second (first) player, respectively.

In this way for the game ${}_1\Gamma$, firstly we can determine the maximum elements for the first player, and for corresponding elements we determine if exist some combinations in the matrix of the second player for that we have Nash equilibria.

The modified algorithm.

For the game ${}_1\Gamma$, we determine the maximum element in each column from the matrix A, i. e. $a_{i_jj}=\max\limits_i\{a_{1j},a_{2j},\dots,a_{mj}\}$, for $\forall j=\overline{1,n}$.

For each element $a_{i_j j}$, $j = \overline{1,n}$ thus obtained, we determine the corresponding elements with the same indices from the matrix $B : b_{i_j j}$, $j = \overline{1,n}$.

For each of these pairs $a_{i_j j}, b_{i_j j}$, $\left(j = \overline{1,n}\right)$ we determine if these values can be the payoffs for players for some Nash equilibria.

Thus if $\forall k \in X_2 \backslash \{j\}\ \exists b_{ik} : b_{ik} \leqslant b_{i_j j}$, then the pair $a_{i_j j}, b_{i_j j}$ can be the payoffs for players for some Nash equilibria in the game ${}_1\Gamma$; consider this pair $a_{i^* j^*}, b_{i^* j^*}$.

It is possible that for the pair $a_{i^* j^*}, b_{i^* j^*}$ there are many Nash equilibria.

If we wish to determine how many Nash equilibria there are in the game ${}_1\Gamma$ for the pair $a_{i^* j^*}, b_{i^* j^*}$ we determine the number of elements which there are in each column $k \in X_2 \backslash \{j\}$ from the matrix B for that $b_{ik} \leqslant b_{i^* j^*}$. Denote by n_j, $j = \overline{1,n}$ the number of elements b_{ij} from the column j for that $b_{ij} \leqslant b_{i^* j^*}$, and for j^* we have $n_{(j^*)} = 1$.

Then the number of Nash equilibria for that the players will have the payoff $a_{i^* j^*}$ and $b_{i^* j^*}$, respectively, can be determined by:

$$N_{j^*} = n_1 \cdot n_2 \cdot \ldots \cdot n_{(j^*-1)} \cdot 1 \cdot n_{(j^*+1)} \cdot \ldots n_n, \tag{4}$$

And the number of all Nash equilibria in the game ${}_1\Gamma$ can be determined by: $N = \sum\limits_j N_j$.

If the pair of elements $a_{i_j j}, b_{_{i_j j}}$ can be the payoffs of the players for some Nash equilibrium in the informational extended game ${}_1\Gamma$, then j will be the strategy for the second player. And because $\overline{X_1} \neq X_1$, we have to determine the strategy for the first player, for which the elements $a_{i^* j^*}, b_{i^* j^*}$ will correspond to one Nash equilibrium.

In this way we determine the elements $b_{i_1 1}, b_{i_2 2}, \ldots, b_{i_j j}, \ldots, b_{i_n n}$, for that $b_{i_k k} \leqslant b_{i_j j}$, $\forall k \in X_2 \backslash \{j\}$.

Then using the indices of the rows of these elements, we can determine the strategy for the first player by:

$$i' = (i_1 - 1)\, m^{n-1} + (i_2 - 1)\, m^{n-2} + \ldots + (i_j - 1)\, m^{n-j} + \ldots + (i_n - 1)\, m^0 + 1. \tag{5}$$

So, the pair i', j is Nash equilibrium for the informational extended game ${}_1\Gamma$: $(i', j) \in NE\,({}_1\Gamma)$.

Similarly, for the game ${}_2\Gamma$, we can determine the strategy for the second player by:

$$j' = (j_1 - 1)\, n^{m-1} + (j_2 - 1)\, n^{m-2} + \ldots + (j_i - 1)\, n^{m-i} + \ldots + (j_m - 1)\, n^0 + 1, \tag{6}$$

where the indices j_i $\left(i = \overline{1,m}\right)$ are determined by the indices of columns of the elements $b_{ij_i} = \max\limits_j \{b_{i1}, b_{i2}, \ldots, b_{in}\}$, $\forall i = \overline{1,m}$.

Example 7.

$$A = \begin{pmatrix} \underline{\mathbf{9}} & 2 & 6 & 0 \\ 2 & \underline{\mathbf{7}} & 7 & 2 \\ 5 & 4 & \underline{\mathbf{9}} & \underline{\mathbf{5}} \\ 3 & 5 & 4 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 3 & 5 & 3 & \underline{\mathbf{9}} \\ \underline{\mathbf{8}} & 2 & 5 & 7 \\ \underline{\mathbf{7}} & 5 & 4 & 1 \\ 2 & 3 & 1 & \underline{\mathbf{4}} \end{pmatrix}.$$

For this game $NE(\Gamma) = \emptyset$. For the informational extended games ${}_1\Gamma$, ${}_2\Gamma$ the extended matrices will have the dimension $[256 \times 4]$ and $[4 \times 256]$, respectively.

For the game ${}_1\Gamma$ we determine the maximum elements in each column from the matrix A, and for the corresponding elements we determine if there are some combinations in the matrix B such that the pair $(a_{i^*j^*}, b_{i^*j^*})$ will be the payoffs for the players.

So, the pair $(a_{11}, b_{11}) = (9, 3)$ will be the payoffs for the players, and the strategy for the second player will be $j^* = 1$.

We determine the combination of elements for which we have NE in the game ${}_1\Gamma : (b_{11}, b_{22}, b_{13}, b_{34}) = (3, 2, 3, 1)$, for that

$i' = (i_1 - 1)\, 4^3 + (i_2 - 1)\, 4^2 + (i_3 - 1)\, 4^1 + (i_4 - 1)\, 4^0 + 1 = 0 + 1 \cdot 4^2 + 0 + 2 \cdot 4^0 + 1 = 19$, so $(19, 1) \in NE\,({}_1\Gamma)$.

For the pair $(a_{11}, b_{11}) = (9, 3)$ we have $\{(19, 1), (31, 1), (51, 1), (63, 1)\} \in NE\,({}_1\Gamma)$.

Similarly, for the pair $(a_{33}, b_{33}) = (9, 4)$ we obtain
$\{(27, 3), (28, 3), (59, 3), (60, 3), (219, 3), (220, 3), (251, 3), (252, 3)\} \in NE\,({}_1\Gamma)$;

for the pair $(a_{22}, b_{22}) = (7, 2)$ we obtain $\{(223, 2)\} \in NE\,({}_1\Gamma)$.

Thus, in the game ${}_1\Gamma$ there are 13 Nash equilibria.

Similarly, for the game ${}_2\Gamma$ we can determine the set of Nash equilibria.

In this case for the pair $(a_{31}, b_{31}) = (5, 7)$ we obtain the follow Nash equilibria: $(3, 65)$, $(3, 66)$, $(3, 67)$, $(3, 68)$, $(3, 113)$, $(3, 114)$, $(3, 115)$, $(3, 116)$, $(3, 193)$, $(3, 194)$, $(3, 195)$, $(3, 196)$, $(3, 241)$, $(3, 242)$, $(3, 243)$, $(3, 244)$ in the game ${}_2\Gamma$.

Thus, in the game ${}_2\Gamma$ there are 16 Nash equilibria.

References

[1] N.S. Kukushkin, V.V. Morozov, "Teoria neantagonisticeskih igr", Moscova, 1984, (ru), pag 46-51.

[2] Novac Ludmila, "Existenţa situaţiilor Nash de echilibru în jocurile bimatriciale informaţional extinse", Analele ştiinţifice, Facultatea de Matematică şi Informatică, USM, vol. 4, Chişinău 2002, pag.66-71.

[3] Hâncu Boris, Novac Ludmila "Informational aspects in the Game Theory", Annals of the Tiberiu Popoviciu Seminar of Functional Equations, Approximation and Convexity, Volume 3, Cluj-Napoca 2005, p. 25-34.

[4] Novac Ludmila, "Informational extended games", Second conference of the Matemathical Society of the Republic of Moldova (dedicated to the 40th Annyversary of the foundation of the Institute of Matemathics and Computer Science of ASM), Communications, Chishinau, 2004, pag.232-234.

[5] Novac Ludmila, "Existenţa situaţiilor Nash de echilibru în jocurile bimatriciale informaţional extinse", Analele ATIC, USM, vol. 4, Chişinău 2004, pag.32-35.

Department of Mathematics and Informatics, Moldova State University, A. Mateevici str. 60,, MD 2009 Chişinău, Moldova

E-mail address: Novac-Ludmila@yandex.ru

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

DEPENDENT TYPES IN MATHEMATICAL THEORY OF PROGRAMMING

VALERIE NOVITZKÁ, ANITA VERBOVÁ

Abstract. In our approach we consider programming as logical reasoning over type theory of a given solved problem. In our paper we follow our work with describing dependent type theory categorically. We introduce dependent types as families of types indexed by terms and we provide rules of dependent type calculus. We describe indexing in terms of special functors, fibrations along display maps over category of type contexts.

1. INTRODUCTION

In our research we consider programming as logical reasoning over type theory of a given solved problem. We built a category of corresponding type theory and logical system over the type theory by fibration, i.e. by a special functor [4] enabling indexing and substitution. In our previous work we presented how to construct in this manner Church's type theory (ChTT) [7] and polymorphic type theory (PTT) [9] categorically and in [8] we built first order logical system over these type theories.

We started our approach with the concept of many-typed signature $\Sigma = (T, F)$ consisting of a finite set T of basic types $\sigma, \tau, \ldots$ needed for a solved problem and a finite family F of operations of the form $f : \sigma_1, \ldots, \sigma_n \to \tau$. From basic types we can construct Church's types using constructors for product $(\sigma \times \tau)$, coproduct $(\sigma + \tau)$ and function $(\sigma \to \tau)$ types and we defined ChTT by classifying category $Cl(\Sigma)$ over Σ consisting of type contexts (variable declarations) $\Gamma = (v : \sigma_1, \ldots, v_n : \sigma_n)$ as category objects and tuples of terms $(t_1, \ldots, t_m) : \Gamma \to \Delta$ as category morphisms, where $\Gamma \vdash t_i : \tau_i$ denotes a term t_i of type τ_i for $i = 1, \ldots, m$ with free variables declared in Γ.

Then we constructed PTT over higher-order signature $(\bar{\Sigma}, (\Sigma_k))$. PTT enables type variables $\alpha, \beta, \ldots$ and kinds $K, L, \ldots$ of types that are enclosed in kind signature $\bar{\Sigma} = (\mathcal{K}, \mathcal{F})$ of kinds and functions. For every kind $k \in \mathcal{K}$ a signature Σ_k

Received by the editors: May 25, 2007.

2000 *Mathematics Subject Classification.* 18A15, 18D30.

1998 *CR Categories and Descriptors.* F.4.1 [**Mathematical logic and formal languages**]: Mathematical logic – *λ-calculus and related systems*; G.2.0 [**Mathematics of computing**]: Discrete mathematics – *General* .

consists of $\bar{\Sigma}$-terms $\alpha_1 : K_1, \ldots, \alpha_m : K_m \vdash \sigma : \texttt{Type}$. Then we constructed PTT by split polymorphic fibration

$$\begin{array}{c} Cl(\bar{\Sigma}, (\Sigma_k)) \\ \Big\downarrow \\ Cl(\bar{\Sigma}) \end{array}$$

with generic object `Type` in $Cl(\bar{\Sigma})$.

In this paper we follow our approach with defining another types, dependent types that have been frequently used in computer science. There are several other approach to capture type dependency, we mention here only contextual categories [15] and D-categories [1]. We prefer fibrations because they enable to express indexing and substituting by display maps and we investigate how to describe DTT categorically in the sense of our previous research.

2. DEPENDENT TYPES

Dependent types are families of types indexed by terms [11]. They offer a degree of precision in describing program behaviours that goes far beyond the other typing features. In dependent type theory (DTT) a term variable $x : \sigma$ can occur in another type $\tau(x) : \texttt{Type}$. As an example we assume a type $IntList$ of lists of integers with operations

$$\begin{array}{ll} nil : & IntList \\ append : & Int, IntList \to IntList \\ head : & IntList \to Int \\ tail : & IntList \to IntList \\ isempty : & IntList \to Bool \end{array}$$

where Int and $Bool$ are types of integers and boolean values, respectively. In DTT we can refine the type $IntList$ to a family of types $IntList(n)$, the types of lists with n elements, where $n : Nat$ is a natural number. In such a manner we form a dependency of the type $IntList(n)$ on the type Nat. To express this dependency between arguments of operations and the types of their results, we consider e.g. that the type of operation $append$ is a function $append : Int, IntList(n) \to IntList(succ(n))$, where $succ$ is an operation of type Nat. It is clear that after appending an element to a list of n elements we get a list of $n + 1$ elements. So we capture in types the dependency between the value of an argument $n : Nat$ on one side and the type $IntList(n)$ and result type $IntList(succ(n))$ on the other side. Such types do not exist in ChTT and PTT. They can be considered to be similar as I-indexed collection $X = (X_i)_{i \in I}$ of sets, which can be written as

$$i : I \vdash X_i : Set$$

where $\vdash I : Set$, i.e. I is an (index) set.

The types of operations for list of integers of length n can be written using a new constructor, *dependent product* $\prod$ as follows:

$$\begin{array}{ll} nil: & IntList(0) \\ append: & \prod n : Nat.(Int, IntList(n)) \to IntList(succ(n)) \\ head: & \prod n : Nat.IntList(succ(n)) \to Int \\ tail: & \prod n : Nat.IntList(succ(n)) \to IntList(n) \end{array}$$

The types of the operations $nil, append$ and $tail$ tell us how many elements are in their results and that $head$ and $tail$ operations demand non-empty lists as arguments. Now we do not need the operation $isempty$ because we can see whether the number of list n is 0. So dependent function types

$$\prod x : \sigma.\tau$$

are more precise form of function types $\sigma \to \tau$ of ChTT. The dependent product constructor binds a variable x representing the argument of the function so that we can mention it in the result type τ.

We can build also higher-level list manipulating operations with similarly refined types. For example, we can define a *new operation*, e.g. sorting function

$$sort : \prod n : Nat.IntList(n) \to IntList(n)$$

that returns a sorted list of the same length as the input. We can also construct *new terms*, e.g.

$$\begin{array}{ll} append3 = & \lambda n : Nat.\lambda i : Int.\lambda l.IntList(n). \\ & append(succ(succ(n))\ i \\ & (append(succ(n))\ i\ (append(succ(n))\ i\ (append(n\ i\ l)) \end{array}$$

which appends three integers in an integer list of type $IntList(n)$ of n elements and returns a list of $n + 3$ elements, i.e. of type $IntList(succ(succ(succ(n))))$.

Dependent types are widely used in computer science, e.g. in the description of digital systems we deal with types of bit vectors of a specific length $n : Nat$, i.e. types $BoolVec(n) = Bool^n$ that can be represented as n-tuples of boolean constants $true, false : Bool$ (or more conveniently $0, 1 : Bool$). The type $BoolVec(n)$ depends on $n : Nat$ [4]. Dependent type theory is often called Martin-Löf type theory [6] but his dependent type calculus contains also a type of all types that leads to Girard's paradox [2]. Dependent type theory is used not only for foundational reasoning [3, 5] but also as a basis for proof tools [10].

3. DEPENDENT TYPE CALCULUS

In this section we describe the syntax of dependent type calculus. In our considerations let Σ be a many-typed signature containing basic types. Because in

dependent types can occur terms, types cannot be introduced separately, so recursion is required. Constructors for dependent types are:

- $\prod x : \sigma.\tau(x)$, i.e. *dependent product* of type $\tau(x)$, where term variable x ranges over type σ;
- $\sum x : \sigma.\tau(x)$, i.e. *dependent sum* of type $\tau(x)$, where term variable x ranges over type σ;
- $Eq_\sigma(x, x')$, i.e. the type of σ-equality for variables x, x' ranging over σ. Equality types are called *identity types*.

A dependent product is a collection of functions $(f)_\sigma$, i.e. indexed by σ, such that for every $i : \sigma$

$$f(i) : \tau[i/x]$$

is of type τ where occurrences of variable x are replaced by $i : \sigma$. A dependent sum is a set of pairs (i, j), where $i : \sigma$ and $j : [i/x]$. The substitution $[i/x]$ in type τ is typical for dependent type theory. Dependent products generalise exponents and dependent sums generalise Cartesian products of ChTT.

If x, x' are variables of the same type, the associated equality type is

$$Eq_\sigma(x, x') = \begin{cases} \{*\} & \text{if } x = x' \\ \emptyset & \text{otherwise} \end{cases}$$

where $\{*\}$is singleton.

We use *type context* $\Gamma = (x_1 : \sigma_1, \ldots, x_n : \sigma)$ denoting a finite sequence of typed variables as in ChTT and PTT but we add the following property: every type σ_{i+1} is a well-formed type in the previous context $\Gamma' = (x_1 : \sigma_1, \ldots, x_i : \sigma_i)$, i.e.

$$x_1 : \sigma_1, \ldots, x_i : \sigma_i \ \vdash \ \sigma_{i+1} : \texttt{Type}$$

From this definition it follows that every free variable $y : \sigma_{i+1}$ must already have been declared in Γ', i.e. it must be one of $x_1, \ldots, x_i$.

Example 1: The well-formed context may be e.g.

$$\Gamma = (n : Nat, l : IntList(n))$$

but

$$\Delta = (n : Nat, z : Array(n, m))$$

is not well-formed because m is not declared.

□

A *sequent of* DTT may have one of the following forms:

$$\Gamma \ \vdash \ \sigma : \texttt{Type} \qquad (1)$$
$$\Gamma \ \vdash \ t : \sigma \qquad (2)$$
$$\Gamma \ \vdash \ t = s : \sigma \qquad (3)$$
$$\Gamma \ \vdash \ \sigma = \tau : \texttt{Type} \qquad (4)$$

The sequent (1) denotes dependent type σ in the context Γ, (2) denotes a term t of type σ in context Γ and (3) expresses equality (conversion) of terms. In (4) is described the conversion of types, because terms may occur in types. From categorical point of view types are equal if they are inhabited by the same terms.

Basic rules for dependent type theory are:

$$\frac{\Gamma \ \vdash \ \sigma : \mathtt{Type}}{\Gamma, x : \sigma \ \vdash \ x : \sigma} \ (proj) \qquad \frac{\Gamma \ \vdash \ t : \sigma \quad \Gamma, x : \sigma, \Delta \ \vdash \ \tau}{\Gamma, \Delta[\tau/x] \ \vdash \ \tau[t/x]} \ (subst)$$

$$\frac{\Gamma, x : \sigma, y : \sigma, \Delta \ \vdash \ \tau}{\Gamma, x : \sigma, \Delta[x/y] \ \vdash \ \tau[x/y]} \ (contr) \ \frac{\Gamma \ \vdash \ \sigma : \mathtt{Type} \quad \Gamma \ \vdash \ \tau}{\Gamma, x : \sigma \ \vdash \ \tau} \ (weak)$$

$$\frac{\Gamma, x : \sigma, y : \tau, \Delta \ \vdash \ \tau}{\Gamma, y : \tau, x : \sigma, \Delta \ \vdash \ \tau} \ (exchange)$$

where τ is an arbitrary expression occurable on the right side of sequent and x is not free in τ. For singleton, i.e. *unit type* 1 we introduce the following rules

$$\frac{}{\vdash \ 1 : \mathtt{Type}} \qquad \frac{}{\vdash \ \langle\,\rangle : 1} \qquad \frac{\Gamma \ \vdash \ t : 1}{\Gamma \ \vdash \ t = \langle\,\rangle : 1}$$

We can form dependent product $\prod$, dependent coproduct $\sum$ and equality type by the following rules:

$$\frac{\Gamma, x : \sigma \ \vdash \ \tau : \mathtt{Type}}{\Gamma \ \vdash \ \prod x : \sigma.\tau : \mathtt{Type}} \qquad \frac{\Gamma, x : \sigma \ \vdash \ \tau : \mathtt{Type}}{\Gamma \ \vdash \ \sum x : \sigma.\tau : \mathtt{Type}}$$

$$\frac{\Gamma \ \vdash \ \sigma : \mathtt{Type}}{\Gamma, x : \sigma, x' : \sigma \ \vdash \ Eq_\sigma(x, x') : \mathtt{Type}}$$

These type constructors change the context. The variable $x : \sigma$ becomes bound in $\prod x : \sigma.\tau$ and $\sum x : \sigma.\tau$. Term substitution can be defined by

$$\begin{array}{ll} (\prod x : \sigma.\tau)[s/y] = & \prod x : \sigma[s/y].\tau[s/y] \\ (\sum x : \sigma.\tau)[s/y] = & \sum x : \sigma[s/y].\tau[s/y] \\ Eq_\sigma(x, x')[s/y] = & Eq_{\sigma[s/y]}(x[s/y], x'[s/y]) \end{array}$$

where in the first two lines we assume that y is different from x and x is not free in s.

Associated rules for terms are the following:

$$\frac{\Gamma, x : \sigma \ \vdash \ t : \tau}{\Gamma \ \vdash \ \lambda x : \sigma.t : \prod x : \sigma.\tau} \ (abstraction)$$

$$\frac{\Gamma \ \vdash \ t : \prod x : \sigma.\tau}{\Gamma \ \vdash \ t \ s : \tau[s/x]} \ (application)$$

$$\frac{\Gamma \ \vdash \ \sigma : \mathtt{Type} \quad \Gamma, x : \sigma \ \vdash \ \tau : \mathtt{Type}}{\Gamma, x : \sigma, y : \tau \ \vdash \ \langle x, y \rangle : \sum x : \sigma.\tau} \ (pairing)$$

$$\frac{\Gamma \ \vdash \ \rho : \texttt{Type} \ \Gamma, x : \sigma, y : \tau \ \vdash \ s : \rho}{\Gamma, z : \sum x : \sigma.\tau \ \vdash \ (\texttt{unp} \ z \ \texttt{as} \ \langle x, y \rangle \ in \ s) : \rho} \ (unpairing)$$

$$\frac{\Gamma \ \vdash \ \sigma : \texttt{Type}}{\Gamma, x : \sigma \ \vdash \ \texttt{refl}_\sigma(x) : Eq_\sigma(x, x)} \ (reflexivity)$$

$$\frac{\Gamma, x : \sigma, x' : \sigma, \Delta \ \vdash \ \rho : \texttt{Type} \qquad \Gamma, x : \sigma, \Delta[x/x'] \ \vdash \ s : \rho[x/x']}{\Gamma, x : \sigma, x' : \sigma, z : Eq_\sigma(x, x'), \Delta \ \vdash \ (s \ \texttt{with} \ x = x' \ \texttt{via} \ z) : \rho}$$

where `unp` is unpairing operator for sum types similar as in [7], $\texttt{refl}_\sigma$ is reflexivity combinator for equality and s `with` $x' = x$ `via` z denotes the elimination term for dependent equality types.

4. ENCLOSING DEPENDENT TYPES IN CATEGORY

We start the categorical investigation of type dependency. We use a distinguished class of morphisms, *display maps* [14] in a category of contexts. A display map $\varphi : (X_i)_{i \in I} \to I$ in set theoretical sense is a mapping from a family $(X_i)_{i \in I}$ of sets to the set I. Then every set $X_i = \varphi^{-1}(i)$ is indexed by the element $i \in I$. Such indexing is equivalent with obvious pointwise indexing but it has a big advantage if considering categories of contexts. Every indexed set X_i can be regarded as a *fibre* subcategory over an object i and φ displays the (total) category $(X_i)_{i \in I}$ over a base category I.

Before we construct DTT categorically, we must consider the following fact: since terms can occur in types we may have conversions between types. But then it is possible to have conversions between contexts (componentwise). Therefore we do not consider contexts Γ as objects of classifying category, but equivalence classes $[\Gamma]$ of contexts w.r.t. conversion. But for notation simplicity we use in the following Γ for $[\Gamma]$.

We assume a fixed dependent type calculus over a signature Σ and we form the classifying category of contexts $ClD(\Sigma)$ for DTT that contains:

- as category objects equivalence classes $\Gamma, \Delta, \ldots$;

- as category morphisms $\Gamma \to \Delta$ (where $\Delta = (y_1 : \tau_1, \ldots, y_m : \tau_m)$, y_i are free in

τ_i) n-tuples $(t_1, \ldots, t_n)$ of terms t_i such that

$$\Gamma \ \vdash \ t_i : \tau_i[t_1/y_1, \ldots, t_{i-1}/y_{i-1}]$$

We denote this n-tuple of terms by $\vec{t} : \Gamma \to \Delta$ and call it a *context morphism*. Explicit substitution in types is typical for DTT and is not in ChTT and PTT. These substitutions are performed simultaneously.

Identity $\Gamma \to \Gamma$ for every context $\Gamma = (x_1 : \sigma_1, \ldots, x_n : \sigma_n)$ is the n-tuple $(x_1, \ldots, x_n)$ of variables.

Composition of morphisms

$$\Gamma \xrightarrow{(t_1,\ldots,t_m)} \Delta \xrightarrow{(s_1,\ldots,s_k)} \Theta$$

where

$$\begin{aligned}\Gamma = &\quad (x_1:\sigma_1,\ldots,x_n:\sigma_n)\\ \Delta = &\quad (y_1:\tau_1,\ldots,y_m:\tau_m)\\ \Theta = &\quad (z_1:\rho_1,\ldots,z_k:\rho_k)\end{aligned}$$

and

$$\begin{aligned}&\Gamma \vdash t_i : \tau_i[t_1/y_1,\ldots,t_{i-1}/y_{i-1}]\\ &\Delta \vdash s_j : \rho_j[s_1/z_1,\ldots,s_{j-1}/z_{j-1}]\end{aligned}$$

is the k-tuple $(u_1,\ldots,u_k) : \Gamma \to \Theta$ with components

$$u_j = s_j[\vec{t}/\vec{y}] = s_j[t_1/y_1,\ldots,t_m/y_m]$$

for $i = 1,\ldots,m$ and $j = 1,\ldots,k$, that are well-typed, i.e.

$$\begin{aligned}\Gamma \;\vdash\quad & u_j : \rho_j[s_1/z_1,\ldots,s_{j-1}/z_{j-1}]\ [\vec{t}/\vec{y}]\\ &= \rho_j[s_1/z_1,\ldots,s_{j-1}/z_{j-1}]\end{aligned}$$

To prove associativity is not so simple and the method of the proof can be found in [14, 12]. Now we can say that $ClD(\Sigma)$ constructed above is a category.

In ChTT and PTT classifying categories have finite products, empty type context is terminal object and concatenation of contexts yields binary products. In DTT is easy to see that empty context again yields a terminal object. But concatenation of contexts does not yield products, but rather dependent sums.

Example 2: Let $(x:\sigma, y:\tau)$ be a type context of two types, where x may occur in τ. A context morphism $\Gamma \to (x:\sigma, y:\tau)$ does not correspond to two morphisms

$$\Gamma \to (x:\sigma) \qquad \Gamma \to (y:\tau)$$

but to the following two morphisms

$$t : \Gamma \to (x:\sigma) \qquad s : \Gamma \to (y:\tau[t/x]) \qquad (*)$$

This dependent pairing property can be described by the existence of pullbacks in the category $ClD(\Sigma)$ along display maps:

$$(\Gamma, z:\rho) \xrightarrow{\varphi} \Gamma.$$

Explicitly, for $\Gamma = (x_1 : \sigma_1,\ldots,x_n : \sigma_n)$ the map $\varphi : (\Gamma, z : \rho) \to \Gamma$ is the n-tuple $(x_1,\ldots,x_n)$ of variables in Γ. Display map is some kind of (dependent) projection because all variables declared in Γ may occur free in ρ. This situation is illustrated in Figure 1, where display maps are closed under pullback (t,s), where $(t,s) : \Gamma \to (x:\sigma, y:\tau)$ is a context morphism and t and s are as in $(*)$.

□

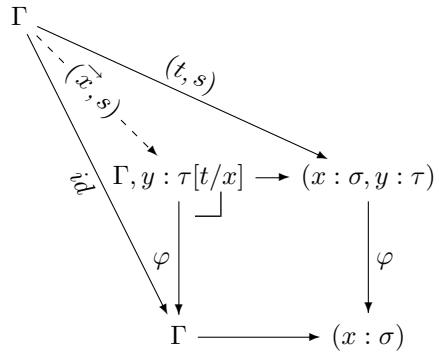


Figure 1: Display map closed under pullback

We denote by $\mathcal{D}$ the collection of display maps

$$\varphi : (\Gamma, x : \sigma) \to \Gamma$$

in $ClD(\Sigma)$ induced by types $\Gamma \vdash \sigma : \texttt{Type}$ in contexts. We construct the *arrow category* $\mathcal{D}^{\to}$ consisting of

- display maps φ as objects and

- pairs of morphisms $(u, g) : \varphi \to \psi$ as category morphisms, where u, g are as in the commutative diagram in Figure 2.

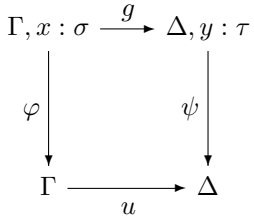


Figure 2: Morphisms in $\mathcal{D}^{\to}$

Maps in $\mathcal{D}$ form a split fibration over $ClD(\Sigma)$ in Figure 3.

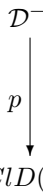


Figure 3: Dependent types fibration

where p assigns to every display map $\varphi : \Gamma, x : \sigma \to \Gamma$ the codomain context, $p(\varphi) = \Gamma$. Substitution functor $\varphi^* : \mathcal{D}^{\to}_{\Gamma} \to \mathcal{D}^{\to}_{\Gamma, x:\sigma}$ along a display map $\varphi : (\Gamma, x : \sigma) \to \Gamma$ in this fibration is functor between corresponding fibre subcategories over corresponding contexts. It is *weakening* because it moves a type $\Gamma \vdash \tau : \texttt{Type}$ to a bigger context $\Gamma, x : \sigma \vdash \tau : \texttt{Type}$ as in the pullback in Figure 4.

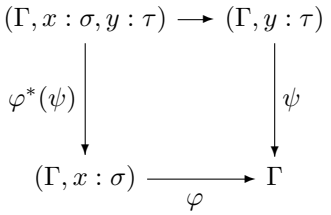
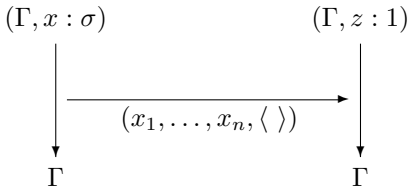


Figure 4: Substitution functor φ^*

A unit type $1 : \tau$ corresponds to a terminal object functor $\mathbf{1} : ClD(\Sigma) \to \mathcal{D}^{\to}$ for the fibration of display maps. This functor is defined by

$$\Gamma \mapsto \begin{pmatrix} \Gamma, z : 1 \\ \downarrow \\ \Gamma \end{pmatrix}$$

Then for arbitrary display map $\varphi : (\Gamma, x : \sigma) \to \Gamma$ there is precisely one pair of morphisms



where $x_1, \ldots, x_n$ are variables declared in Γ.

Dependent products $\prod$ and sums $\sum$ correspond to the fibration in Figure 2 having Cartesian products and sums along display maps $\varphi : (\Gamma, x : \sigma) \to \Gamma$ in $ClD(\Sigma)$. This means that dependent products correspond to right adjoints of weakening functors φ^* along display maps and dependent sums $\sum$ correspond to left adjoints along display maps. For dependent products and sums must hold Beck-Chevalley conditions [4], i.e. for any morphism $t : \Gamma, x : \sigma \to \Gamma$ in $ClD(\Sigma)$ and every pair of reindexing (substitution) functors $\varphi^*, \varphi^{\#} : \mathcal{D}^{\to}_{\Gamma} \to \mathcal{D}^{\to}_{\Gamma, x:\sigma}$ between fibres, natural transformation is an identity.

We can say that the codomain fibration in Figure 2 from display maps arrow category to classifying category characterizes categorically DTT. Every object Γ in classifying category indexes a fibre subcategory $\mathcal{D}^{\to}_{\Gamma}$. Dependent type constructors $1, \prod$ and $\sum$ are defined by adjoints to substitution functors in total category $\mathcal{D}^{\to}$.

5. Conclusion

In this contribution we presented dependent type theory categorically as fibration from arrow category of display maps to classifying category consisting of dependent type contexts. We can say now that we have integrated categorical approach for representing ChTT, PTT and DTT. Over these type theories we

construct logical system also as fibration similarly as in [9], i.e. in the case of DTT it will be double fibration over classifying category. In the following research we would like to extend this approach also to higher-order dependent type theory, based on polymorphic and dependent types.

This work is supported by the grant VEGA 1/2181/05: Mathematical theory of programming and its application in the methods of stochastic programming

References

[1] Th.Ehrhard: A categorical semantics of constructions, In: Logic in Computer Science, IEEE, Computer Science Press, 1988, pp.264-273
[2] J.-Y.Girard: Interprétation fonctionelle et élimination des coupures dans l'arithmetique d'order supéieur, PhD.Thesis, Université Paris, 1972
[3] M.Hofmann: Syntax and semantics of dependent types, Semantics and Logic of Computation, Cambridge Univ.Press, 1997
[4] B.Jacobs: Categorical logic and type theory, Elsevier, Amsterdam, 1999
[5] Z.Luo: Computation and reasoning: A type theory for computer science, Monographs on Computer Science, Oxford Univ.Press, 1994
[6] P.Martin-Löf: Intuitionistic type theory, Bibliopolis, Napoli, 1984
[7] V.Novitzká: Church's types in logical reasoning on programming, Acta Electronica et Informatica, Vol.6,No.2,2006, Košice, pp.27-31
[8] V.Novitzká, D.Mihályi, V.Slodičák: Categorical logic over Church's types, Proc. 6th Scient.Conf. Electronic Computers and Informatics ECI'2006, Košice-Herľany, September 2006, pp.122-129
[9] V.Novitzká, D.Mihályi: Polymorphic type theory for categorical logic, Acta Electrotechnica et Informatica, Kosice, To appear, 2007
[10] S.Owre et al: Formal verification for fault-tolerant architectures: Prolegomena to the design of PVS, IEEE Trans. on Softw.Eng., Vol.21, No.2, 1995, pp.107-125
[11] B.C.Pierce: Types and programming languages, MIT, 2002
[12] A.M.Pitts: Categorical logic, In: S.Abramsky, D.M.Gabbai, T.S.E.Maibaum (eds.): Handbook of Logic in Computer Science, Vol.6, Oxford Univ.Press, 1995
[13] P.Taylor: Recursive domains, indexed category theory and polymorphism, PhD.Thesis, Univ.Cambridge, 1986
[14] P.Taylor: Practical foundations of mathematics, Cambridge Univ.Press, 1999
[15] Th.Streicher: Semantics of type theory. Correctness, completeness and independence results, Progress in Theoretical ComputerScience, Birkhauser, Boston, 1991

Faculty of Electrical Engineering and Informatics, Technical University of Košice
E-mail address: valerie.novitzka@tuke.sk, anita.verbova@tuke.sk

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

AN ANALYSIS OF DISTANCE METRICS FOR CLUSTERING BASED IMPROVEMENT OF SYSTEMS DESIGN

ISTVÁN GERGELY CZIBULA AND GABRIELA ŞERBAN

Abstract. *Clustering* is the process of grouping a set of objects into classes of similar objects. *Refactoring* is the process of improving the design of software systems.It improves the internal structure of the system, but without altering the external behavior of the code ([5]). In [1] we have proposed a new approach for improving systems design using *clustering*. The aim of this paper is to analyze the influence of distance metrics for clustering based improvement of systems design. We are focussing on identifying the most suitable distance metric. The study is made on the clustering based approach developed in [1].

Keywords: software engineering, refactoring, clustering, distance metrics.

1. Introduction

The software systems, during their life cycle, are faced with new requirements. These new requirements imply updates in the software systems structure, that have to be done quickly, due to tight schedules which appear in real life software development process. That is why continuous restructurings of the code are needed, otherwise the system becomes difficult to understand and change, and therefore it is often costly to maintain.

Refactoring is a solution adopted by most modern software development methodologies (extreme programming and other agile methodologies), in order to keep the software structure clean and easy to maintain. Thus, refactoring becomes an integral part of the software development cycle: developers alternate between adding new tests and functionality and refactoring the code to improve its internal consistency and clarity.

In [5], Fowler defines refactoring as "the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure. It is a disciplined way to clean up code that minimizes

Received by the editors: January 15, 2007.

2000 *Mathematics Subject Classification.* 68N99, 62H30.

1998 *CR Categories and Descriptors.* D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement –*Restructuring, reverse engineering, and reengineering*; I.5.3 [**Computing Methodologies**]: Pattern Recognition – *Clustering*.

the chances of introducing bugs". Refactoring is viewed as a way to improve the design of the code after it has been written. Software developers have to identify parts of code having a negative impact on the system's maintainability, and apply appropriate refactorings in order to remove the so called "bad-smells" ([10]).

1.1. **Related Work.** There are various approaches in the literature in the field of *refactoring*, but just a few of them use *clustering* in order to restructure programs. In [2] a clustering based approach for program restructuring at the functional level is presented. This approach focuses on automated support for identifying ill-structured or low cohesive functions. The paper [11] presents a quantitative approach based on clustering techniques for software architecture restructuring and reengineering as well for software architecture recovery. It focuses on system decomposition into subsystems.

We have developed, in [1], a *k-means* based clustering approach for identifying refactorings in order to improve the structure of software systems. For this purpose, *kRED* (k-means for REfactorings Determination) algorithm is introduced.

In this paper we study the influence of distance metrics on the results obtained by *kRED* algorithm. We intend to identify the most suitable distance metric using the open source case study JHotDraw ([14]).

The rest of the paper is structured as follows. The approach (*CARD*) proposed in [1] for determining refactorings using a clustering technique is presented in Section 2. Section 3 provides an experimental evaluation of *CARD*, based on different distance metrics and using the open source case study JHotDraw ([14]). Some conclusions and further work are outlined in Section 4.

2. Refactorings Determination using a Clustering Approach

In this section we briefly describe the clustering approach (*CARD*), introduced in [1], that aims at finding adequate refactorings in order to improve the structure of software systems.

CARD approach consists of three steps:

- **Data collection** - The existent software system is analyzed in order to extract from it the relevant entities: classes, methods, attributes and the existent relationships between them.
- **Grouping** - The set of entities extracted at the previous step are regrouped in clusters using a *k-means* based clustering algorithm, *kRED* ([1]). The goal of this step is to obtain an improved structure of the existing software system.
- **Refactorings extraction** - The newly obtained software structure is compared with the original software structure in order to provide a list of refactorings which transform the original structure into an improved one.

2.1. **Theoretical model.** We have introduced in [1] a theoretical model on which *CARD* approach is based on. In the following we will briefly describe this model.

Let $S = \{s_1, s_2, ..., s_n\}$ be a software system, where $s_i, 1 \leq i \leq n$ can be an application class, a method from a class or an attribute from a class.

We will consider that:

- $Class(S) = \{C_1, C_2, \ldots, C_l\}$, $Class(S) \subset S$, is the set of applications classes in the initial structure of the software system S.
- Each application class C_i $(1 \leq i \leq l)$ is a set of methods and attributes, i.e., $C_i = \{m_{i1}, m_{i2}, \ldots, m_{ip_i}, a_{i1}, a_{i2}, \ldots, a_{ir_i}\}$, $1 \leq p_i \leq n$, $1 \leq r_i \leq n$, where m_{ij} $(\forall j,\ 1 \leq j \leq p_i)$ are methods and a_{ik} $(\forall k,\ 1 \leq k \leq r_i)$ are attributes from C_i.
- $Meth(S) = \bigcup_{i=1}^{l} \bigcup_{j=1}^{p_i} m_{ij}$, $Meth(S) \subset S$, is the set of methods from all the application classes of the software system S.
- $Attr(S) = \bigcup_{i=1}^{l} \bigcup_{j=1}^{r_i} a_{ij}$, $Attr(S) \subset S$, is the set of attributes from the application classes of the software system S.

Based on the above notations, the software system S is defined as in Equation (1):

$$S = Class(S) \bigcup Meth(S) \bigcup Attr(S). \tag{1}$$

As described above, at the **Grouping** step of our approach, the software system S has to be re-grouped. In our view, this re-grouping is represented as a ***partition*** of S.

Definition 1. *([1])**Partition of a software system** S.*
*The set $\mathcal{K} = \{K_1, K_2, ..., K_v\}$ is called a **partition** of the software system $S = \{s_1, s_2, \ldots, s_n\}$ iff*

- $1 \leq v \leq n$;
- $K_i \subseteq S, K_i \neq \emptyset, \forall i,\ 1 \leq i \leq v$;
- $S = \bigcup_{i=1}^{v} K_i$ *and* $K_i \cap K_j = \emptyset,\ \forall i, j, 1 \leq i, j \leq v, i \neq j$.

In the following, we will refer to K_i as the i-th *cluster* of $\mathcal{K}$, to $\mathcal{K}$ as a *set of clusters* and to an element s_i from S as an *entity*.

A cluster K_i from the partition $\mathcal{K}$ represents an application class in the new structure of the software system.

2.2. **kRED algorithm.** In [1], based on the theoretical model described in Subsection 2.1, a *k-means* based clustering algorithm (*kRED*) is introduced. The

algorithm is used in the *Grouping* step of *CARD*, and aims at identifying a **partition** of a software system S that corresponds to an improved structure of it.

kRED is a vector space model based *clustering* algorithm, that is used in order to re-group entities from the software system.

In *CARD* approach ([1]), the objects to be clustered are the entities from the software system S, i.e., $\mathcal{O} = \{s_1, s_2, \ldots, s_n\}$ and the attribute set is the set of application classes from the software system S, $\mathcal{A} = \{C_1, C_2, \ldots, C_l\}$.

The focus is to group similar entities from S in order to obtain high cohesive groups (clusters), that is why is considered the dissimilarity degree between the entities and the application classes C from S, $\forall C,\ C \in Class(S)$.

Consequently, each entity s_i $(1 \leq i \leq n)$ from the software system S is characterized by a l-dimensional vector: $(s_{i1}, s_{i2}, \ldots, s_{il})$, where s_{ij} $(\forall j,\ 1 \leq j \leq l)$ is computed as follows ([1]):

$$s_{ij} = \begin{cases} -\frac{|p(s_i) \cap p(C_j)|}{|p(s_i) \cup p(C_j)|} & \text{if } \ p(s_i) \cap p(C_j) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}, \tag{2}$$

where, for a given entity $e \in S$, $p(e)$ defines a set of relevant properties of e, expressed as:

- If $e \in Attr(S)$ (e is an attribute) then $p(e)$ consists of: the attribute itself, the application class where the attribute is defined, and all methods from $Meth(S)$ that access the attribute.
- If $e \in Meth(S)$ (e is a method) then $p(e)$ consists of: the method itself, the application class where the method is defined, and all attributes from $Attr(S)$ accessed by the method.
- If $e \in Class(S)$ (e is an application class) then $p(e)$ consists of: the application class itself, and all attributes and methods defined in the class.

A more detailed justification of the vector space model choice is given in [1].

As in a vector space model based clustering ([8]), we consider the *distance* between two entities s_i and s_j from the software system $\mathcal{S}$ as a measure of dissimilarity between their corresponding vectors. We will consider in our study three possible distance metrics between methods:

- *Euclidian Distance.* The distance between s_i and s_j is expressed as:

$$d_E(s_i, s_j) = \sqrt{\sum_{k=1}^{l} (s_{ik} - s_{jk})^2} \tag{3}$$

- *Manhattan Distance.* The distance between s_i and s_j is expressed as:

$$d_M(s_i, s_j) = \sum_{k=1}^{l} |s_{ik} - s_{jk}| \tag{4}$$

- *Cosine Distance.* The distance between s_i and s_j is expressed as:

$$d_C(s_i, s_j) = \frac{\sqrt{\sum_{k=1}^{l} s_{ik}^2} \cdot \sqrt{\sum_{k=1}^{l} s_{jk}^2}}{\sum_{k=1}^{l} (s_{ik} \cdot s_{jk})}. \tag{5}$$

The main idea of *kRED* algorithm is the following ([1]):

(i) The initial number of clusters is the number l of application classes from the software system $\mathcal{S}$.
(ii) The initial centroids are chosen as the application classes from $\mathcal{S}$.
(iii) As in the classical *k-means* approach, the clusters (centroids) are recalculated, i.e., each object is assigned to the closest cluster (centroid).
(iv) Step (iii) is repeatedly performed until two consecutive iterations remain unchanged, or the number of steps performed exceeds the maximum number of iterations allowed.

In the following we intend to analyze the influence of the distance metrics described above on the results obtained by *kRED* algorithm.

3. Experimental Evaluation

For our analysis, we will consider two evaluations, which are described in Subsections 3.1 and 3.2. We will evaluate the results obtained by applying *kRED* algorithm for the distance metrics defined in Subsection 2.2.

3.1. **Code Refactoring Examples.**

3.1.1. *Example 1.* We aim at studying how the *Move Method* refactoring is obtained after applying *kRED* algorithm, for the analyzed distance metrics.

Let us consider the Java code example shown in Figure 1.

Analyzing the code presented in Figure 1, it is obvious that the method **methodB1()** has to belong to **class_A**, because it uses features of **class_A** only. Thus, the refactoring *Move Method* should be applied to this method.

We have applied *kRED* algorithm ([1]), for *Euclidian distance* and *Manhattan distance*, and the *Move Method* refactoring for **methodB1()** was determined. The two obtained clusters are:

- Cluster 1:
  {**Class_A, methodA1(), methodA2(), methodA3(), methodB1(), attributeA1, attributeA2**}.
- Cluster 2:
  {**Class_B, methodB2(), methodB3(), attributeB1, attributeB2**}.

```
public class Class_A {
  public static int attributeA1;
  public static int attributeA2;

  public static void methodA1(){
      attributeA1 = 0;
      methodA2();
  }

  public static void methodA2(){
          attributeA2 = 0;
          attributeA1 = 0;
    }

  public static void methodA3(){
          attributeA2 = 0;
          attributeA1 = 0;
          methodA1();
          methodA2();
  }
}
```

```
public class Class_B {
  private static int attributeB1;
  private static int attributeB2;

  public static void methodB1(){
          Class_A.attributeA1=0;
          Class_A.attributeA2=0;
          Class_A.methodA1();
    }

  public static void methodB2(){
          attributeB1=0;
          attributeB2=0;
    }

  public static void methodB3(){
          attributeB1=0;
          methodB1();
          methodB2();
  }
}
```

FIGURE 1. Code example for *Move Method* refactoring

The first cluster corresponds to application class **Class_A** and the second cluster corresponds to application class **Class_B** in the new structure of the system.

For *Cosine distance*, the *Move Method* refactoring for **methodB1()** is not identified.

3.1.2. *Example 2.* We aim to analyze how the *Move Attribute* refactoring is obtained after applying *kRED* algorithm, for the studied distance metrics. Let us consider the Java code example shown in Figure 2.

Analyzing the code presented in Figure 2, it is obvious that the attribute **attributeA1** has to belong to **class_B**, because is mostly used by methods from **class_B**. Thus, the refactoring *Move Attribute* should be applied to this attribute.

We have applied *kRED* algorithm for *Euclidian distance* and *Manhattan distance*, and the *Move Attribute* refactoring for **attributeA1** was determined.

For each distance metric, the two obtained clusters are:

```
public class Class_A {

  public static int attributeA2;
  public static int attributeA1;

  public static void methodA1() {
    methodA2();
  }

  public static void methodA2() {
    attributeA2 = 0;
    Class_A.attributeA1 = 12;
  }

  public static void methodA3() {
    attributeA2 = 0;
    methodA1();
    methodA2();
  }
}

public class Class_B {

  private static int attributeB1;
  private static int attributeB2;

  public static void methodB1() {
    attributeB1 = 0;
    Class_A.methodA1();
  }

  public static void methodB2() {
    attributeB1 = 0;
    attributeB2 = 0;
    Class_A.attributeA1 = 12;
  }

  public static void methodB3() {
    attributeB1 = 0;
    methodB1();
    methodB2();
    Class_A.attributeA1 = 12;
  }

  public static void methodB4() {
    attributeB1 = 0;
    methodB2();
    Class_A.attributeA1 = 12;
  }
}
```

FIGURE 2. Code example for *Move Attribute* refactoring

- Cluster 1:
  {**Class_A, methodA1(), methodA2(), methodA3(), attributeA2**}.
- Cluster 2:
  {**Class_B, methodB1(), methodB2(), methodB3(), methodB4(), attributeA1, attributeB1, attributeB2**}.

The first cluster corresponds to application class **Class_A** and the second cluster corresponds to application class **Class_B** in the new structure of the system.

For *Cosine distance*, the *Move Attribute* refactoring for **attributeA1** is not identified.

From the examples in Subsection 3.1 we can conclude, experimentally, that *Euclidian distance* and *Manhattan distance* are the most appropriate distance metrics.

3.2. **JHotDraw Case Study.** Our second analysis is made on the open source software JHotDraw, version 5.1 ([14]). It is a Java GUI framework for technical and structured graphics, developed by Erich Gamma and Thomas Eggenschwiler, as a design exercise for using design patterns.

The reason for choosing JHotDraw as a case study is that it is well-known as a good example for the use of design patterns and as a good design.

In order to test the accuracy of *CARD* approach, two measures ACC and $PREC$ were introduced in [1]. These measures indicate how accurate are the results obtained after applying *kRED* algorithm in comparison to the current design of JHotDraw. We assume that $\mathcal{K} = \{K_1, \dots K_p\}$ is a partition reported after applying *kRED* algorithm.

Definition 2. *([1])**ACCuracy of a refactoring technique - ACC.***

Let $\mathcal{T}$ be a refactoring technique.

The accuracy of $\mathcal{T}$ with respect to a partition $\mathcal{K}$ and the software system S, denoted by $ACC(S, \mathcal{K}, \mathcal{T})$, is defined as:

$$ACC(S, \mathcal{K}, \mathcal{T}) = \frac{1}{l} \sum_{i=1}^{l} acc(C_i, \mathcal{K}, \mathcal{T}).$$

$acc(C_i, \mathcal{K}, \mathcal{T}) = \dfrac{\sum\limits_{j \in \mathcal{M}_{C_i}} \dfrac{|C_i \cap K_j|}{|C_i \cup K_j|}}{|\mathcal{M}_{C_i}|}$ *(where $\mathcal{M}_{C_i} = \{j |\ 1 \leq j \leq p,\ |C_i \cap K_j| \neq 0\}$ is the set of clusters from $\mathcal{K}$ that contain elements from the application class C_i), is the accuracy of $\mathcal{T}$ with respect to the application class C_i.*

ACC defines the degree to which the partition $\mathcal{K}$ is similar to S. Based on Definition 2, it can be proved that larger values for ACC indicate better partitions with respect to S, meaning that ACC has to be maximized.

Definition 3. [1] ***PRECision of a refactoring technique - PREC.***

Let $\mathcal{T}$ be a refactoring technique.

The precision of methods dicovery in $\mathcal{T}$ with respect to a partition $\mathcal{K}$ and the software system S, denoted by $PREC(S, \mathcal{K}, \mathcal{T})$, is defined as:

$$PREC(S, \mathcal{K}, \mathcal{T}) = \frac{1}{|Meth(S)|} \sum_{m \in Meth(S)} prec(m, \mathcal{K}, \mathcal{T}).$$

$prec(m, \mathcal{K}, \mathcal{T}) = \begin{cases} 1 & \text{if } m \text{ was placed in the same class as in } S \\ 0 & \text{otherwise} \end{cases}$ *,is the precision of $\mathcal{T}$ with respect to the method m.*

$PREC(S, \mathcal{K}, \mathcal{T})$ defines the percentage of methods from S that were correctly discovered by $\mathcal{T}$ (we say that a method is correctly discovered if it is placed in its original application class). Based on Definition 3, it can be proved that larger values for $PRECM$ indicate better partitions with respect to S, meaning that $PREC$ has to be maximized.

After applying *kRED* algorithm for JHotDraw case study, for the considered distance metrics, we obtain the results described in Table 1.

Distance metric	ACC	PREC
Euclidian distance	**0.9829**	**0.997**
Manhattan distance	**0.9721**	**0.9949**
Cosine distance	**0.9829**	**0.997**

TABLE 1. The measures values.

The results from Table 1 show that the results obtained for *Euclidian distance* and *Cosine distance* are the best, because provide the largest values for ACC and $PREC$.

As a conclusion, from the results obtained in Subsections 3.1 and 3.2 we can conclude, experimentally, that *Euclidian distance* is the most suitable distance metric to be used in *kRED* algorithm.

4. Conclusions and Future Work

We have analyzed in this paper the influence of distance metrics for clustering based improvement of systems design. We have comparatively present the results obtained by *kRED* algorithm ([1]) for different distance metrics. In order to evaluate the obtained results, we have used two quality measures defined in [1].

Based on the obtained results, we can conclude, experimentally, that *Euclidian distance* is the most suitable distance metric to be used in *kRED* algorithm.

Further work can be done in the following directions:

- To apply *kRED* for other case studies, like JEdit ([3]).
- To use other approaches for clustering, such as hierarchical clustering ([8]), search based clustering ([7]), or genetic clustering ([13]).
- To improve the vector space model used for clustering.

References

[1] Czibula, I.G., Serban, G.: Improving Systems Design Using a Clustering Approach. IJCSNS International Journal of Computer Science and Network Security, VOL.6, No.12 (2006) 40–49

[2] Xu, X., Lung, C.H., Zaman, M., Srinivasan, A.: Program Restructuring Through Clustering Technique. In: 4th IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2004), USA (2004) 75–84

[3] jEdit Programmer's Text Editor: http://www.jedit.org (2002)

[4] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
[5] Fowler, M.: Improving the design of existing code. Addison-Wesley, New-York (1999)
[6] Simon, F., Loffler, S., Lewerentz, C.: Distance based cohesion measuring. In Proceedings of the 2nd European Software Measurement Conference (FESMA) 99, Technologisch Instituut Amsterdam (1999)
[7] Doval, D., Mancoridis, S., Mitchell, B.S.: Automatic clustering of software systems using a genetic algorithm. IEEE Proceedings of the 1999 Int. Conf. on Software Tools and Engineering Practice STEP'99 (1999)
[8] Jain, A., Murty, M.N., Flynn, P.: Data clustering: A review. ACM Computing Surveys **31** (1999) 264–323
[9] Bieman, J.M., Kang, B.-K.: Measuring Design-Level Cohesion. In: IEEE Transactions on Software Engineering **24** No. 2 (1998)
[10] McCormick, H., Malveau, R.: Antipatterns: Refactoring Software, Architectures, and Projects in Crises. John Wiley and Sons (1998)
[11] Lung, C.H.: Software Architecture Recovery and Restructuring through Clustering Techniques. ISAW3, Orlando, SUA (1998) 101–104
[12] Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, New Jersey (1998)
[13] Cole, R.M.: Clustering with genetic algorithms. Master's thesis, University of Western Australia (1998)
[14] JHotDraw Project: http://sourceforge.net/projects/jhotdraw (1997)
[15] Chidamber, S., Kemerer, C.: A metrics suite for object oriented design. Im: IEEE Transactions on Softwareengineering **20** No. 6 (1994) 476–493

Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: istvanc@cs.ubbcluj.ro

Department of Computer Science, Babeş-Bolyai University 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: gabis@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

ON EVALUATING SOFTWARE SYSTEMS DESIGN

GABRIELA ŞERBAN AND ISTVÁN GERGELY CZIBULA

Abstract. We have previously introduced in [1, 2] clustering approaches for identifying refactorings in order to improve the structure of software systems. For this purpose, we have defined in [2] a *semi-metric* function in order to express the dissimilarity between the entities from a software system. In this paper we aim at giving a theoretical validation for this *semi-metric*. In other words, we are focussing on proving that this function illustrates the cohesion between the entities from a software system and can be used in order to obtain appropriate refactorings of it.

Keywords: software engineering, refactoring, distance metric, clustering.

1. Introduction

Improving the quality of a software system design is the most important issue during the evolution of object oriented software systems.

Refactoring is the process of improving the design of software systems. It improves the internal structure of the system, but without altering the external behavior of the code ([5]).

During the software development cycle, there is a continuous alternance between adding new tests and functionalities for a software system, and refactoring the code in order to improve its internal consistency and clarity.

We have previously introduced in [1, 2] clustering approaches for identifying refactorings in order to improve the structure of software systems. For this purpose, we have defined in [2] a *semi-metric* function in order to express the dissimilarity between the entities from a software system.

Received by the editors: April, 20.

2000 *Mathematics Subject Classification.* 68N99, 62H30.

1998 *CR Categories and Descriptors.* D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement –*Restructuring, reverse engineering, and reengineering*; I.5.3 [**Computing Methodologies**]: Pattern Recognition – *Clustering*.

The main contribution of this paper is to give a theoretical validation for this *semi-metric* function, and, consequently, a theoretical validation for the clustering approaches.

The rest of the paper is structured as follows. The main aspects related to the clustering approach for systems design improvement (previously introduced in [1, 2]) are exposed in Section 2.1. The theoretical validation of the *semi-metric* distance d used in the clustering approach from [2] is given in Section 3. Section 4 provides an experimental validation of d. Conclusions and further work are outlined in Section 5.

2. Background

2.1. **Clustering approach for refactorings determination.** In this section we briefly describe the clustering approach for improving software systems design (*CARD*) previously introduced in [1, 2].

In [1], a software system S is viewed as a set $S = \{s_1, s_2, \ldots, s_n\}$, where $s_i, 1 \leq i \leq n$, may be an application class, a method from a class, or an attribute from a class. *CARD* consists of three steps:

- **Data collection.** The existent software system is analyzed in order to extract from it the relevant entities: classes, methods, attributes and the existent relationships between them.
- **Grouping.** The set of entities extracted at the previous step are re-grouped in clusters using a partitioning algorithm (like *kRED* algoritm in [1] or *PAMRED* algorithm in [2]). The goal of this step is to obtain an improved structure of the existing software system.
- **Refactorings extraction.** The newly obtained software structure is compared with the original software structure in order to provide a list of refactorings which transform the original structure into an improved one.

A more detailed description of *CARD* is given in [1].

At the **Grouping** step of *CARD*, the software system S has to be re-grouped. This re-grouping is represented as a ***partition*** $\mathcal{K} = \{K_1, K_2, \ldots, K_v\}$ of S. K_i is the i-th *cluster* of $\mathcal{K}$, and an element s_i from S is referred as an *entity*. A cluster K_i from the partition $\mathcal{K}$ represents an application class in the new structure of the software system.

2.2. **A semi-metric dissimilarity function.** In the clustering approaches from [1, 2], the objects to be clustered are the entities from the software system S, i.e.,

$\mathcal{O} = \{s_1, s_2, \ldots, s_n\}$. Our focus is to group similar entities from S in order to obtain high cohesive groups (clusters).

In [1], we have adapted the generic cohesion measure introduced in [6] that is connected with the theory of similarity and dissimilarity. In order to express the dissimilarity degree between any two entities from the software system S, we have considered the distance $d(s_i, s_j)$ between two entities s_i and s_j as expressed in Equation (1) ([1]).

$$d(s_i, s_j) = \begin{cases} 1 - \frac{|p(s_i) \cap p(s_j)|}{|p(s_i) \cup p(s_j)|} & if \quad p(s_i) \cap p(s_j) \neq \emptyset \\ \infty & otherwise \end{cases}, \tag{1}$$

where, for a given entity $e \in S$, $p(e)$ represents a set of relevant properties of e, defined as:

- If e is an attribute, then $p(e)$ consists of: the attribute itself, the application class where the attribute is defined, and all methods from S that access the attribute.
- If e is a method, then $p(e)$ consists of: the method itself, the application class where the method is defined, and all attributes from S accessed by the method.
- If e is a class, then $p(e)$ consists of: the application class itself, and all attributes and methods defined in the class.

We have chosen the distance between two entities as expressed in Equation (1) because it emphasizes the idea of cohesion. As illustrated in [7], "*Cohesion refers to the degree to which module components belong together*".

Based on the definition of distance d given in Equation (1) it can be easily proved that d is a semi-metric function.

3. Theoretical validation

In this section we are focusing on giving a theoretical validation of the *semi-metric* dissimilarity function d described in Subsection 2.2. We aim at proving that our distance, as defined in Equation (1), highlight the concept of cohesion, i.e., entities with low distances are cohesive, whereas entities with higher distances are less cohesive. This theoretical validation of d will give a validation of the clustering approach from [2], also.

Let us consider that e, α and β are three entities from the software system S, $e \neq \alpha \neq \beta$. In Lemma 1 we give a necessary and sufficient condition in order to illustrate that entity e is more distant from entity β than from entity α.

We consider, in the following, the definition of the distance function d given in Equation (1).

Lemma 1. *If e, α and β are three entities from the software system S, and $p(e) \cap p(\alpha) \neq \emptyset$, then*

$$d(e, \alpha) < d(e, \beta) \tag{2}$$

iff

$$\frac{|p(e) \cap p(\alpha)|}{|p(e)| + |p(\alpha)|} > \frac{|p(e) \cap p(\beta)|}{|p(e)| + |p(\beta)|}. \tag{3}$$

Proof.

Since $p(e) \cap p(\alpha) \neq \emptyset$, we have that

$$|p(e) \cap p(\alpha)| > 0. \tag{4}$$

From (4) we can deduce that:

$$d(e, \alpha) = 1 - \frac{|p(e) \cap p(\alpha)|}{|p(e) \cup p(\alpha)|}. \tag{5}$$

First, we prove implication "⇒" from Lemma 1.

Let us assume that Inequality (2) holds. We have to prove that Inequality (3) holds, also.

We have two situations in which Inequality (2) holds:

1. $d(e, \beta) = \infty$.

   This means (from Equation (1)) that entities e and β are unrelated and have no common relevant properties. Consequently we have that $p(e) \cap p(\beta) = \emptyset$. It follows that

$$|p(e) \cap p(\beta)| = 0. \tag{6}$$

   From (6) and (4) we have that Inequality (3) holds. So, implication "⇒" from Lemma 1 is proved.
2. $d(e, \beta) < 1$.

   This means (from Equation (1)) that entities e and β are related and have common relevant properties. Consequently, using (5), it follows that:

$$1 - \frac{|p(\alpha) \cap p(e)|}{|p(\alpha) \cup p(e)|} < 1 - \frac{|p(\beta) \cap p(e)|}{|p(\beta) \cup p(e)|}. \tag{7}$$

From (7) we have that:

$$\frac{|p(\beta) \cap p(e)|}{|p(\beta) \cup p(e)|} < \frac{|p(\alpha) \cap p(e)|}{|p(\alpha) \cup p(e)|}. \tag{8}$$

If A and B are two sets, it is well known that Equality (9) holds.

$$|A \cup B| = |A| + |B| - |A \cap B|. \tag{9}$$

From (8) and (9) it follows that:

$$|p(\beta) \cap p(e)| \cdot (|p(\alpha)| + |p(e)| - |p(\alpha) \cap p(e)|) < \\ |p(\alpha) \cap p(e)| \cdot (|p(\beta)| + |p(e)| - |p(\beta) \cap p(e)|). \tag{10}$$

Using (10) we have that:

$$|p(\beta) \cap p(e)| \cdot (|p(\alpha)| + |p(e)|) < |p(\alpha) \cap p(e)| \cdot (|p(\beta)| + |p(e)|). \tag{11}$$

Inequality (11) implies Inequality (3). So, implication "⇒" from Lemma 1 is proved.

Now, we prove implication "⇐" from Lemma 1.

Let us assume that Inequality (3) holds. We have to prove that Inequality (2) holds, also.

We have two situations in which Inequality (3) holds:

1. $p(\beta) \cap p(e) = \emptyset$.

   This means (from Equation (1)) that entities e and β are unrelated and have no common relevant properties. Consequently we have that $d(e, \beta) = \infty$. It follows that Inequality (2) holds and implication "⇐" from Lemma 1 is proved.
2. $p(\beta) \cap p(e) \neq \emptyset$.

   This means (from Equation (1)) that entities e and β are related and have common relevant properties. Consequently it follows that:

$$d(e, \beta) = 1 - \frac{|p(e) \cap p(\beta)|}{|p(e) \cup p(\beta)|}. \tag{12}$$

From (3) we have that:

$$|p(\alpha) \cap p(e)| \cdot (|p(\beta)| + |p(e)|) > |p(\beta) \cap p(e)| \cdot (|p(\alpha)| + |p(e)|). \tag{13}$$

Consequently, we can deduce that:

$$|p(\alpha) \cap p(e)| \cdot (|p(\beta)| + |p(e)|) - |p(\alpha) \cap p(e)| \cdot |p(\beta) \cap p(e)| > \\ |p(\beta) \cap p(e)| \cdot (|p(\alpha)| + |p(e)|) - |p(\alpha) \cap p(e)| \cdot |p(\beta) \cap p(e)|. \tag{14}$$

From (14) we have that:

$$|p(\alpha) \cap p(e)| \cdot (|p(\beta)| + |p(e)| - |p(\beta) \cap p(e)|) > \\ |p(\beta) \cap p(e)| \cdot (|p(\alpha)| + |p(e)| - |p(\alpha) \cap p(e)|). \tag{15}$$

From (15) and (9) it follows that:

$$|p(\alpha) \cap p(e)| \cdot |p(\beta) \cup p(e)| > |p(\beta) \cap p(e)| \cdot |p(\alpha) \cup p(e)| \tag{16}$$

Using (16) we can deduce that:

$$\frac{|p(\alpha) \cap p(e)|}{|p(\alpha) \cup p(e)|} > \frac{|p(\beta) \cap p(e)|}{|p(\beta) \cup p(e)|}. \tag{17}$$

Consequently, we have that:

$$1 - \frac{|p(\alpha) \cap p(e)|}{|p(\alpha) \cup p(e)|} < 1 - \frac{|p(\beta) \cap p(e)|}{|p(\beta) \cup p(e)|}. \tag{18}$$

From (18), (5) and (12) it follows that Inequality (2) holds and implication "$\Leftarrow$" from Lemma 1 is proved.

As both implications "$\Rightarrow$" and "$\Leftarrow$" from Lemma 1 were proved, Lemma 1 is also proved.

Let us consider that α and β are two entities of the software system S that are situated in different application classes, and e is an entity of S that has to be disposed in one of the application classes (corresponding to α or β).

In this situation, Inequality (3) from Lemma 1 expresses that the number of elements that α has in common with e with respect to the total number of elements from α and e is greater than the number of elements that β has in common with

e with respect to the total number of elements from β and e. This is very likely to express that e is more cohesive with α than with β.

Intuitively, condition (3) is very probable a necessary and suficient condition to indicate that e belongs to the same application class with α and not with β. This statement cannot be rigurously proved, because the decision that an entity to be disposed in an application class or another is very complex and cannot be quantified using rigurous mathematical measures. In practice, the developers decide wheather or not an entity is disposed into an application class, and the decision can be a subjective one. Still, in Subsection 3.1 we give an experimental justification for this statement.

Consequently, we can consider that Proposition 1 is valid.

Proposition 1. *If α is an entity of S situated in application class A and β is an entity of S that is situated in application class B ($B \neq A$), and e is an entity of S that has to be disposed in one of the application classes A or B, then e belongs to A and does not belong to B iff Inequality (3) holds.*

From Lemma 1 and Proposition 1 results the mathematical validation of the semi-metric distance function d, i.e., the facts that:

(1) Entities with low distances are cohesive, whereas entities with higher distances are less cohesive.
(2) The distances between less cohesive entities are greater than the distances between cohesive entities.

This theoretical validation of d is given in Lemma 2.

Lemma 2. *If α is an entity of S situated in application class A and β is an entity of S that is situated in application class B ($B \neq A$), and e is an entity of S that has to be disposed in one of the application classes A or B, then e belongs to A and does not belong to B iff $d(e, \alpha) < d(e, \beta)$.*

From Lemma 2 we can conclude that the decision about putting an entity e from S into an application class or another is based on the distances between e and the entities from the corresponding application classes.

Consequently, a clustering approach that uses the semi-metric d for expressing the dissimilarity between the entities from the software system is very appropriate, and can be used in order to recondition the class structure of a software system, because it expresses the cohesion between the entities from it.

3.1. **Example.** Let us consider the software system S given by the Java code example shown in Figure 1.

```
public class Class_A {
  public static int attributeA1;
  public static int attributeA2;

  public static void methodA1(){
      attributeA1 = 0;
      methodA2();
  }

  public static void methodA2(){
          attributeA2 = 0;
          attributeA1 = 0;
    }

  public static void methodA3(){
          attributeA2 = 0;
          attributeA1 = 0;
          methodA1();
          methodA2();
  }
}
```

```
public class Class_B {
  private static int attributeB1;
  private static int attributeB2;

  public static void methodB1(){
          Class_A.attributeA1=0;
          Class_A.attributeA2=0;
          Class_A.methodA1();
    }

  public static void methodB2(){
          attributeB1=0;
          attributeB2=0;
    }

  public static void methodB3(){
          attributeB1=0;
          methodB1();
          methodB2();
  }
}
```

FIGURE 1. Code example for *Move Method* refactoring

Analyzing the code presented in Figure 1, it is obvious that the method **methodB1()** has to belong to **class_A**, because it uses features of **class_A** only. This means, according to Proposition 1, that Inequality (3) holds for $e =$ **methodB1()**, $\forall \alpha \in$ **class_A**, and $\forall \beta \in$ **class_B**.

Analyzing the code from Figure 1 we can observe that all other entities e from both classes **class_A** and **class_B**, excepting **methodB1()** are correctly disposed in their application classes *App*. This means, according to Proposition 1, that Inequality (3) holds for e, $\forall \alpha \in App$, and $\forall \beta \in C$ ($C \in Class(S)$, $C \neq App$).

We have verified Proposition 1 $\forall e, \alpha, \beta \in S$ that satisfy its hypothesis and we have concluded that the proposition is valid.

For lack of space, we will illustrate in Table 1 the validity of Proposition 1 for e=**methodB1()** and in Table 2 the validity of Proposition 1 for e=**methodB2()**. We aim at illustrating that:

(i) **methodB1()** is more cohesive with entities from **class_A**, consequently the *Move Method* refactoring **methodB1()** from **class_B** to **class_A** will be determined by the clustering approach from [2].

(ii) **methodB2()** is more cohesive with entities from **class_B**, consequently its position in **class_B** is correct.

Items (i) and (ii) are expressed in Tables 1 and 2 by giving in column **Class** the class in which entity $e \in \{$**methodB1()**, **methodB2()**$\}$ should be disposed. The correct application class in which **methodB1()** should be disposed is **class_A** and the correct application class in which **methodB2()** should be disposed is **class_B**.

The results illustrated in Tables 1 and 2 validate experimentally Proposition 1 for the considered software system, according to the considerations in this subsection.

4. Experimental Validation

An experimental validation of the *semi-metric d* (Subsection 2.2) is given in [2]. In [2] we have introduced, based on the *semi-metric d*, a *k-medoids* like clustering algorithm (*PAMRED*) for identifying refactorings in order to improve the design of a software system. *PAMRED* algorithm can be used in the **Grouping** step of *CARD*.

PAMRED algorithm is evaluated on the open source case study JHotDraw ([8]) and a comparison with previous related approaches is also given. This comparison illustrates that *CARD* with *PAMRED* algorithm is better than other similar approaches existing in the literature in the field of refactoring.

5. Conclusions and Future Work

We have given in this paper a theoretical validation of the *semi-metric* function d previously introduced in [1] and used in the clustering approach introduced in [2]. In fact, we have validated our previous approach from [2], approach that can be used to determine refactorings in order to improve the structure of a software system. As a future work we intend:

- To give theoretical validation for other distance functions that were used in our previous clustering approaches for refactorings determination ([1]).
- To develop other distance metrics to be used in clustering approaches for refactorings determination and to give theoretical validations for them.

e	α	β	$\frac{\lvert p(e)\cap p(\alpha)\rvert}{\lvert p(e)\rvert+\lvert p(\alpha)\rvert} - \frac{\lvert p(e)\cap p(\beta)\rvert}{\lvert p(e)\rvert+\lvert p(\beta)\rvert}$	**Class**
methodB1()	methodA1()	methodB2()	0.11111111	Class_A
methodB1()	methodA1()	methodB3()	0.02222222	Class_A
methodB1()	methodA1()	attributeB1()	0.11111111	Class_A
methodB1()	methodA1()	attributeB2()	0.097222224	Class_A
methodB1()	methodA1()	Class_B	0.055555552	Class_A
methodB1()	methodA2()	methodB2()	0.11111111	Class_A
methodB1()	methodA2()	methodB3()	0.02222222	Class_A
methodB1()	methodA2()	attributeB1	0.11111111	Class_A
methodB1()	methodA2()	attributeB2	0.097222224	Class_A
methodB1()	methodA2()	Class_B	0.055555552	Class_A
methodB1()	methodA3()	methodB2()	0.16161618	Class_A
methodB1()	methodA3()	methodB3()	0.07272728	Class_A
methodB1()	methodA3()	attributeB1	0.16161618	Class_A
methodB1()	methodA3()	attributeB2	0.14772728	Class_A
methodB1()	methodA3()	Class_B	0.10606061	Class_A
methodB1()	methodB2()	attributeA1	-0.16161618	Class_A
methodB1()	methodB2()	attributeA2	-0.08888889	Class_A
methodB1()	methodB2()	Class_A	-0.1388889	Class_A
methodB1()	methodB3()	attributeA1	-0.07272728	Class_A
methodB1()	methodB3()	attributeA2	0.0	Class_A
methodB1()	methodB3()	Class_A	-0.049999997	Class_A
methodB1()	attributeA1()	attributeB1	0.16161618	Class_A
methodB1()	attributeA1()	attributeB2	0.14772728	Class_A
methodB1()	attributeA1()	Class_B	0.10606061	Class_A
methodB1()	attributeA2()	attributeB1	0.08888889	Class_A
methodB1()	attributeA2()	attributeB2	0.075	Class_A
methodB1()	attributeA2()	Class_B	0.03333333	Class_A
methodB1()	attributeB1()	Class_A	-0.1388889	Class_A
methodB1()	attributeB2()	Class_A	-0.125	Class_A
methodB1()	Class_A	Class_B	0.08333333	Class_A

TABLE 1. Validation of Proposition 1 for entity e=methodB1().

REFERENCES

[1] Czibula, I.G., Serban, G.: Improving Systems Design Using a Clustering Approach. IJCSNS International Journal of Computer Science and Network Security, VOL.6, No.12 (2006) 40–49

e	α	β	$\frac{\lvert p(e)\cap p(\alpha)\rvert}{\lvert p(e)\rvert+\lvert p(\alpha)\rvert} - \frac{\lvert p(e)\cap p(\beta)\rvert}{\lvert p(e)\rvert+\lvert p(\beta)\rvert}$	**Class**
methodB2()	methodA1()	methodB1()	-0.11111111	Class_B
methodB2()	methodA1()	methodB3()	-0.33333334	Class_B
methodB2()	methodA1()	attributeB1	-0.375	Class_B
methodB2()	methodA1()	attributeB2	-0.42857143	Class_B
methodB2()	methodA1()	Class_B	-0.36363637	Class_B
methodB2()	methodA2	methodB1()	-0.11111111	Class_B
methodB2()	methodA2	methodB3()	-0.33333334	Class_B
methodB2()	methodA2	attributeB1	-0.375	Class_B
methodB2()	methodA2	attributeB2	-0.42857143	Class_B
methodB2()	methodA2	Class_B	-0.36363637	Class_B
methodB2()	methodA3	methodB1()	-0.11111111	Class_B
methodB2()	methodA3	methodB3()	-0.33333334	Class_B
methodB2()	methodA3	attributeB1	-0.375	Class_B
methodB2()	methodA3	attributeB2	-0.42857143	Class_B
methodB2()	methodA3	Class_B	-0.36363637	Class_B
methodB2()	methodB1	attributeA1	0.11111111	Class_B
methodB2()	methodB1	attributeA2	0.11111111	Class_B
methodB2()	methodB1	Class_A	0.11111111	Class_B
methodB2()	methodB3	attributeA1	0.33333334	Class_B
methodB2()	methodB3	attributeA2	0.33333334	Class_B
methodB2()	methodB3	Class_A	0.33333334	Class_B
methodB2()	attributeA1	attributeB1	-0.375	Class_B
methodB2()	attributeA1	attributeB2	-0.42857143	Class_B
methodB2()	attributeA1	Class_B	-0.36363637	Class_B
methodB2()	attributeA2	attributeB1	-0.375	Class_B
methodB2()	attributeA2	attributeB2	-0.42857143	Class_B
methodB2()	attributeA2	Class_B	-0.36363637	Class_B
methodB2()	attributeB1	Class_A	0.375	Class_B
methodB2()	attributeB2	Class_A	0.42857143	Class_B
methodB2()	Class_A	Class_B	-0.36363637	Class_B

TABLE 2. Validation of Proposition 1 for entity e=methodB2().

[2] Serban, G., Czibula, I.G.: A New Clustering Approach for Systems Designs Improvement. 2007 International Conference on Software Engineering Theory and Practice, SETP-07, Orlando, USA (2007) to appear

[3] Simon, F., Steinbruckner, F., Lewerentz, C.: Metrics based refactoring. In: Proc. European Conf. Software Maintenance and Reengineering. IEEE Computer Society Press (2001) 30-38

[4] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
[5] Fowler, M.: Improving the design of existing code. Addison-Wesley, New-York (1999)
[6] Simon, F., Loffler, S., Lewerentz, C.: Distance based cohesion measuring. In Proceedings of the 2nd European Software Measurement Conference (FESMA) 99, Technologisch Instituut Amsterdam (1999)
[7] Bieman, J.M., Kang, B.-K.: Measuring Design-Level Cohesion. In: IEEE Transactions on Software Engineering **24** No. 2 (1998)
[8] JHotDraw Project: http://sourceforge.net/projects/jhotdraw (1997)

Department of Computer Science, Babeş-Bolyai University 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: gabis@cs.ubbcluj.ro

Department of Computer Science, Babeş-Bolyai University, 1, M. Kogalniceanu Street, Cluj-Napoca, Romania,
E-mail address: istvanc@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

EFFICIENT DATA SYNCHRONIZATION FOR MOBILE WIRELESS MEDICAL USERS

ADRIAN SERGIU DARABANT AND HOREA TODORAN

Abstract. In order to take the appropriate decisions as quick as possible, medical doctors need fast access to various pieces of information on their patients. The required information should be accurate, up-to-date, and available on the spot. Even more, after finishing his/her investigation, the medical doctor should be able to immediately forward the relevant results to other medical personnel (doctors, nurses, hospital administration). All this could be implemented in a mobile solution using handheld devices. The goal of our paper is to present a pilot implementation of a medical database system with dynamic and efficient data synchronization using wireless technologies.
Key Words: mobile applications, wireless data synchronization, personal digital assistants, medical applications.

1. Introduction

Being able to perform virtually the same tasks as desktop computers or laptops, Personal Digital Assistants (PDAs) are constantly increasing their attractiveness. Although originally created to run simple applications like electronic diaries, address books or planning calendars, the handheld devices eventually evolved into real pocket computers capable of undertaking more complex tasks, from word processing and spreadsheet editing to multimedia authoring. Current models support data transfer over communication networks via the common wireless protocols (infrared, bluetooth, WiFi, GPRS), therefore providing access to convenient services, including web browsing, messaging, email, and so on.

Besides the *lower power consumption* (the battery life is approximately two times longer than in the case of laptops), the most important advantages of the handheld computers over other mobile devices achieving similar performance consist in their higher *portability* and *mobility*. PDAs are much lighter than laptops or TabletPCs (approximately 100-200g), fit into the jacket's pocket (wearable), can be hold into one hand and operated with the other (handhelds), and can be operated even on the move. Furthermore, they are very easy to use and prove

Received by the editors: August 1, 2007.

2000 *Mathematics Subject Classification.* 68P15.

1998 *CR Categories and Descriptors.* D.2.11 [**Software Engineering**]: Software architectures –*Domain-specific architectures*.

economic viability [1], having much of the computing capability and storage capacity of laptops at a fraction of the cost (some authors even called them "equity computers"- e.g. Andrew Trotter in [2]). As a consequence, PDAs are more and more exploited in various fields, including mobile business ([3]), mobile education, medicine, and leisure.

Nonetheless, there are also inconveniences when using handheld devices. The most significant are related to the *small size of the screen*, which confines the amount of information displayed or requires the intensive use of navigation bars. *Data input* is more difficult than in the case of desktop computers or laptops, since the keyboard and the mouse are very small (if present). Even the stylus pens are rather narrow, therefore requiring accurate operation on the screen pad. Handhelds have relatively limited storage capabilities, are difficult to upgrade and much less robust than TabletPCs, laptops, and desktop computers. Taking all these restrictions into account, software producers must design applications running on PDAs more carefully than those for the other types of PCs. As a good practice example, the Windows Mobile for PocketPC family includes scale down versions of the Microsoft Windows operating systems, very similar to the desktop versions, though adapted in terms of minimal requirements (memory, processor speed) and visual elements (windows, menus, lists, buttons) to the characteristics of PocketPCs. An exhaustive list of general design requirements for Windows Mobile-based PocketPC applications is given in [4].

2. Problem Formulation

Medical doctors need to be efficient when consulting their patients in the daily routine, in terms of both saving time and taking the appropriate professional decisions. A crucial prerequisite for achieving a high level of efficiency is the quick access to a whole range of information about the current patient: medical history, results of previous medical investigations, opinions of other specialists on the case, and so on. Moreover, the medical doctor should be able to easily disseminate the results of his/her own investigations on the patient.

The traditional solution is to use paper-based patient files that have to be carried by the medical doctor or by the accompanying staff to the patient's bed. The more information required for a certain investigation, the bulkier the dossier and the more difficult the search for relevant data.

A modern alternative is to use a computer-based solution, with a piece of equipment that is small enough to fit easily into the jacket's pocket, and with sufficient computing and communication power to rapidly bring patient's data on demand from a central database onto the device.

Although there are still situations where it is feasible for medical doctors to carry the paper-based patient files with them, these do not always provide the most up-to-date information, which is essential when taking the decisions. For instance, the most recent results of laboratory investigations might have not yet

been recorded in the dossier. Furthermore, even if the raw data has already been recorded, it has not yet been interpreted according to medical procedures.

In this paper, we propose an innovative software architecture on mobile devices with communication capacities, solution that can be easily used with any exiting medical data management software. The application not only facilitates the management of various sets of information (doctors, patients, investigations, and so on), using the PDA, but also synchronizes the data between the mobile device and the hospital's database server.

We implemented the proposed framework on mobile assistants (PDAs) with cellular phone and/or wireless 802.11 capabilities [6, 7] in a pilot project that has been experimentally linked with the management system of a hospital. With the aid of the new framework a hospital that has many premises is able to manage, at any moment, updated information about its patients, even coming from far locations. It is also able to keep in contact with general practitioners who, after visiting their patients at home, can immediately send the results of their investigations to the specialists.

The novelty of our system consists in using an incremental data synchronization mechanism based on timestamps, as described in section 4.1 below. The system architecture (see section 3 for details) also ensures a high degree of independence between the mobile system and the hospital's data management system, which is crucial in case of temporary failure of one of the components. Optimized network traffic is achieved by means of a data compression solution, illustrated in section 4.3 of the present paper.

2.1. **Data flow.** The system relies on the existence of the hospital's central database server to which medical staff (doctors, nurses, laboratory assistants), as well as administrative personnel shall have access anytime and from any location within hospital's premises. They not only frequently ask for updated information concerning the patients, but also send new information to the database server (e.g. results of their investigations).

Medical doctors and their assisting staff query the hospital's database to get useful information that could help them conducting their investigations on the patient. As a result of these investigations, the medical staff attains new information that should be inserted into the database for future use. Various pieces of the recorded information could also be used by the administrative staff, for instance to estimate the cost of the treatment or to plan the use of equipment.The Fig. 1 below depicts the typical information flow within the medical architecture we are trying to model.

The general practitioner is in our scenario a mobile user that travels with his/her PDA. Usually, traveling means either visiting the patients in the hospital, or receiving them in his/her office, but general practitioners might as well visit the patients at their residence in certain situations. Given this *volatile* environment,

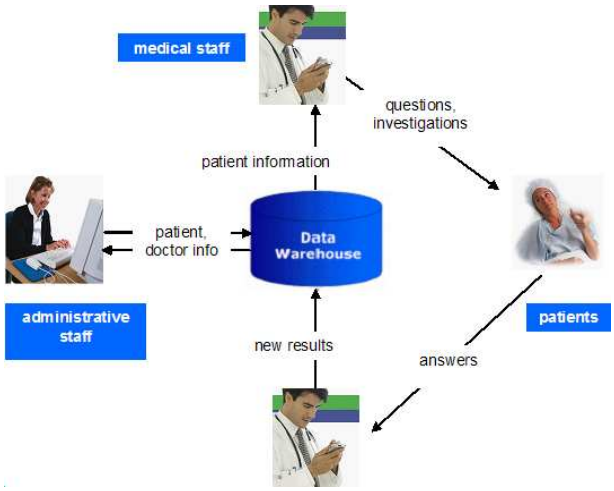


FIGURE 1. Medical information flow

we would like the medical doctor to have as much information as possible about the patients he/she is consulting at a given time. Furthermore, a medical doctor could get assignments for visiting patients directly. The patient's consultation request at the clinic is either directly pushed by the software system on the general practitioner's mobile device or collected in the course of the next data synchronization between the central system and the mobile device.

During a consultation the general practitioner usually evaluates the patient's state using various medical analysis methods: blood sampling, temperature, blood pressure, etc. The results of all these measurements consist in various data sets that the general practitioner would normally write in the patient's record. Using our mobile application - *MobMed* - the general practitioner no longer needs to fill in paper reports and records about the patient. All he has to do is to input the various data sets he obtained into the database, using the friendly interface we designed. Data is temporarily stored in its own portable device (PDA) and then synchronized with the central warehouse of the hospital. Virtually, the general practitioner has a small sized tool capable of storing patient's records, recalling history data about the patients and therefore helping the medical doctor to establish an early potential diagnosis.

3. System Architecture and Services

In the following paragraphs we present the proposed system architecture that has been already implemented in a pilot project called *MobMed*. One of our major requirements was to build a system that is as least intrusive as possible in the

existing *hospital* or clinic *management software* (HMS). With this goal in mind, we needed to link the mobile system facilities to an existing software solution in such a way that no major adjustments are required for the already implemented solution.

We used the following model to implement the mobile architecture on the top of an existing hospital management software, which is running on an MS Windows-like operating system and storing data in an SQL Database Server. We present this particular architectural model here in order to show the degree of independence between the mobile system and the existing management solution.

Even if it is a challenging task, the integration between any existing management software (HMS) and *MobMed* is always possible with little or no intrusion into the HMS at all. In order to implement the *MobMed* solution we suppose that the clinic (hospital) already has at list an Intranet system, possibly connected to the Internet by some highly secure connection. We further suppose that, in most of the cases, the internal network of the hospital is protected by a firewall that separates the Intranet from the Internet.

In Fig. 2 below the general practitioners are running MobMed on Pocket PC devices with incorporated GSM phones or wireless radio cards (WiFi). The choice of GPRS/UMTS or wireless is conditioned by the necessary connection persistence. If the connection is desired at any particular moment, the best choice would be GPRS/UMTS, both requiring a GSM line. For connections only in hospital's perimeter the use of the 802.11 standard and WiFi hotspots might be a convenient alternative.

Since we want to give access to the mobile general practitioners to the database server where patient, diagnosis and consultation schedules data are stored, some features of the Microsoft based platforms can be used. First of all, the SQL Database Server, in its 2000 and 2005 variants, offers support for mobile subscribers to data publications.

The *MobMed* server runs SQL Server and provides for data synchronization with the mobile devices. The synchronization procedure uses the HTTP protocol for data transport in order to be easily accessible on highly secured platforms. The mobile devices are running a .NET application platform with Mobile SQL Server 2005 as data storage and synchronization engine. Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile), the "descendant" of Microsoft SQL Server 2000 CE Edition 2.0 (SQL Server 2000), extends the Microsoft enterprising solutions for *line-of-business* and for the management of personal information applied on a device. SQL Server Mobile delivers the functionalities necessary to a relational database, transposed to a lower scale: robust information storage, query preparation, connectivity capacities. Microsoft SQL Server 2005 was projected to support an extended list of mobile devices and Table PCs. The mobile devices include any device that runs Microsoft Windows CE 5.0, Microsoft Mobile Pocket PC 2003,

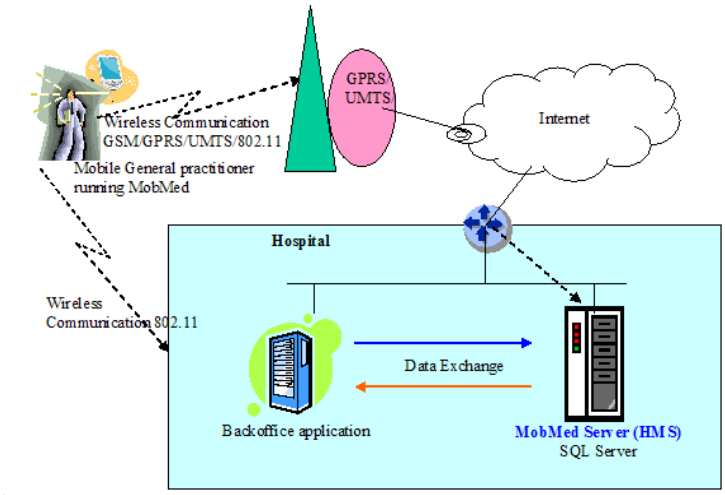


FIGURE 2. The MobMed Architecture and Integration with the Hospital Management Software

Microsoft Mobile Version 5.0 Pocket PC, or Microsoft Mobile Version 5.0 Smart Phone.

Merge replication[1] is an ideal synchronization mechanism when using mobile devices, as it allows the automatic and independent data update on both the mobile device and the server. As soon as the device is connected, the data is synchronized, changes being sent from the client to the server, along with the new information being pushed from the server to the mobile device. Merge Replication needs more server configurations and maintenance than its alternative method (Remote Data Access), but it is more advantageous for applications that involve several mobile devices. Merge Replication also has the capacity of detecting and solving shown up conflicts and of rejoining data from several tables at the same time. It allows instrument survey by using SQL Server and provides more data rejoining options such as the article types and filtering.

Merge Replication is a proper solution when conflict solving is required, when information has to be propelled to and from the desktop or laptop computers and when working with larger databases.

Merge Replication uses some components of the Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile): SQL Server Mobile Database Engine, SQL Server Mobile Client Agent, SQL Server Mobile Server Agent, SQL Server Mobile Replication Provider.

[1]Microsoft TechNet[http://technet.microsoft.com/en-us/library/ms171927.aspx]

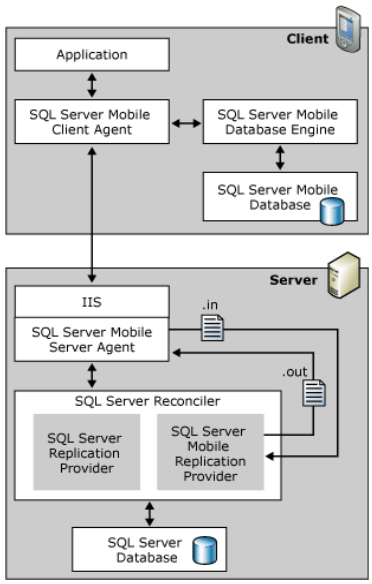


FIGURE 3. Merge Replication Architecture

4. Data Exchange and Synchronization

As already mentioned, the synchronization method uses HTTP as transport protocol. HTTP is used because of its high implementation availability on all platforms and because it allows easier through firewall communication.

4.1. **Merge replication implementation.** The functioning of *merge replication* from Microsoft SQL Server 2005 Mobile Edition implies the following processes:

(1) The data is published on the SQL Server
(2) A *subscriber* is created for the publication
(3) The data in the subscriber is updated
(4) The data is synchronized.

Details on data publication and subscriber registration setup are presented in [8].

When a SQL Server Mobile *subscriber* is synchronized with the SQL Server, all the changes upon data are recovered by the database publication. However, when a SQL Server Mobile subscriber is synchronized for the first time, it can recover the data either directly from the database publication or from the snapshot file. The sealed class *SqlCeReplication* is part of the space names *System.Data.SqlServerCe* and allows the synchronization of SQL Server Mobile databases with the SQL Server databases. It's interface is described in [9].

The typical steps to perform data synchronization when everything is setup are depicted bellow:

```
SqlCeReplication repl = null;
Try{
    repl = new SqlCeReplication();
    repl.InternetUrl = "http://www.myHMSserver.ro/sqlmobile/sqlcesa30.dll";
    repl.InternetLogin = "MyInternetLogin";
    repl.InternetPassword = "<password>";
    repl.Publisher = "MyPublisher";
    repl.PublisherDatabase = "MyPublisherDatabase";
    repl.PublisherLogin = "MyPublisherLogin";
    repl.PublisherPassword = "<password>";
    repl.Publication = "MyPublication";
    repl.Subscriber = "MySubscriber";
    repl.SubscriberConnectionString = "Data Source=MyDatabase.sdf";
    repl.AddSubscription(AddOption.CreateDatabase);
    repl.Synchronize();
} catch (SqlCeException) {     // Error handling/ }
finally   {    repl.Dispose();  }
```

4.2. **Conflict Resolution.** An important aspect in data synchronization with multiple data sources is conflict resolution. Devices might update the same entity locally with different data and then upload changes to the *MobMed* server in a random order. The problem is choosing the correct final value/state of the entity. Automated conflict resolution is usually tightly dependent on the business rules that govern the conflicting entity's type ([10]). SQL Server Mobile detects *client-side* conflicts but does not manipulate their settlement. The conflict information is sent to the Publisher in view of settlement during the following synchronization. Most of the conflicts are settled by the Publisher following the synchronization.

4.3. **Data Compression and Features.** An important aspect when dealing with GSM transports is the cost of the data transfer. As the GPRS/UMTS solutions are still costly, the employment of a data compression mechanism is beneficial. The application has native compression support based on Merge replication compression. Another feature that reduces traffic is column and cell based synchronization. When just a single column of database row has been updated, only that value is sent to the server avoiding thus an entire row transmission.

As already mentioned in the introductory section of this paper, software applications for PDAs have to be carefully designed in order to overcome the limitations of the handheld devices, especially those related to the small size of screen and the difficult use of data input accessories (keyboard, mouse, stylus pen). We certainly took these constraints into account when designing the user interface of the *MobMed* solution.

Following are the main characteristics of the *MobMed* user-friendly interface:

- the navigation bar always displays the name of the topmost window, thus avoiding confusion;
- common menus appear on the leftmost position of the MenuBar in a known order;

- whenever text fields are present on the screen, they are accessible with the Soft Input Panel (SIP) up, thus facilitating data input from mobile devices without keyboard;
- the application maintains the regional and language settings specified by the client.
- in order to ease user's navigation through the successive application's windows, we employed suggestive graphical elements wherever appropriate. It is also the case of the main menu, made of attractive graphical buttons.

5. Conclusions and future work

The current paper proposes a novel multi-tier system to enhance the mobility of medical staff, thus bringing an important efficiency advantage to the hospital/clinic implementing it.

The main functional characteristics of our proposal are related to medical staff having access to up-to-date, complex information on their Pocket PCs, virtually wherever and whenever they need it. This is provided by on demand secure and fast synchronization of the local database from the mobile device with the central data warehouse of the hospital.

The application running on the mobile device (MobMed) has a user-friendly interface, which is designed according to the general requirements described in [4].

As far as the system architecture is concerned, our model ensures application isolation and independence in the case of temporary failure between the three tiers: the mobile device, the data management server, and the client's existing management software. Moreover, *MobMed* successfully deals with important aspects related to data synchronization, like conflict resolution and data compression.

Incremental data synchronization, optimized network traffic, and cvasi-permanent availability are features that make our system valuable and different from other similar implementations.

Future work is intended to building a general framework allowing the smooth integration with any HMS. It should also improve the security of data exchange and the control of mobile agents' navigation from the handheld devices.

References

[1] E. Dieterle, *Handheld devices for ubiquitous learning and analyzing*, 2005 National Educational Computing Conference, Philadelphia, PA, available at [http://center.uoregon.edu/ISTE/uploads/NECC2005/KEY_7287575/Dieterle_NECC2005Dieterle_RP.pdf], visited June 2007.

[2] A. Trotter, *Palm computing moving from the workplace to the classroom*, in Education Week, October 1999.

[3] A.S. Darabant, H. Todoran, *Building an Efficient Architecture for Data Synchronization on Mobile Wireless Agents*, in WSEAS Transactions on Communications, Issue 8, Volume 5, August 2006, ISSN 1109-2742, pp. 1384-1391.

[4] ***, Designed for Windows Mobile *Software Application Handbook for Pocket PCs*, Microsoft Corporation, May 2004.
[5] M.D. Sutton, *Data Synchronization: Which Technology?*, Intel Software Network, [http://www.intel.com/cd/ids/developer/asmo-na/eng/52893.htm], visited June 2007.
[6] ***, *IBM eServer i5 and iSeries System Handbook: IBM i5/OS version 5*, IBM Corporation, October 2004, [http://www.redbooks.ibm.com/redbooks/GA195486/wwhelp/wwhimpl/java/html/wwhelp.htm], visited June 2007.
[7] M. Mallick, *Mobile and Wireless Design Essentials*, Wiley Publishers, 2003.
[8] Microsoft TechNet, emphHow to: Create a Publication and Define Articles (SQL Server Management Studio), Microsoft Corporation, [http://technet.microsoft.com/en-us/library/ms151160.aspx], visited June 2007.
[9] MSDN, *SqlCeReplication Class*, Microsoft Corporation Documentation, [http://msdn2.microsoft.com/en-us/library/system.data.sqlserverce.sqlcereplication.aspx], visited June 2007.
[10] W. Wheeler, *Integrating Wireless Technology in the Enterprise*, First Edition: PDAs, Blackberries, and Mobile Devices, Elsevier Digital Press, 2003.

Babes-Bolyai University, Cluj-Napoca, Romania
E-mail address: dadi@cs.ubbcluj.ro

Babes-Bolyai University, Cluj-Napoca, Romania
E-mail address: dadi@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

THE 3SST RELATIONAL MODEL

ANDREEA SABAU

Abstract. In order to represent spatio-temporal data, many conceptual models have been designed and a part of them have been implemented. This paper presents the relational model of the 3SST conceptual model, as the corresponding implementation of it on a relational database platform. The property of generality invoked during the conceptual modeling operation is inherited by the relational model. The concrete model is able to represent thematic, spatial, temporal, spatio-temporal and event-type objects, as the conceptual model. The spatial objects can be represented as points or objects with shape and the evolution of the spatio-temporal objects can be implemented as discrete or continuous in time, on time instants or time intervals. More than that, different types of spatial, temporal, ST and event-based queries can be performed on represented data. Therefore, the 3SST relational model can be considered the core of a ST data model.

1. Introduction

Spatio-temporal databases (STDB) deal with spatial objects that are changing over time and space. In other words, these objects are characterized by spatial and temporal attributes. Yet, it is important to mention that these are not static objects: the spatial characteristics are changing in time. Another condition to consider a database to be spatio-temporal, is to store not only the last state (or the current state) of objects, but their evolution in time as well. Actually, the temporal characteristic of the stored data consists of this condition [4, 9].

After years during which the spatial and temporal databases were studied and developed independently, the need of using spatial and temporal data in the same application appeared. The first attempts consisted in adding one dimension to the other: including temporal data into a spatial database or adding spatial attributes to the temporal objects. Later, other models joined space and time into one unified spatio-temporal view [11].

There are many domains where the spatio-temporal (ST) data is used today:

Received by the editors: September 1, 2007.

2000 *Mathematics Subject Classification.* 68P05, 68P20.

1998 *CR Categories and Descriptors.* H.2.1 [**Information Systems**]: Database Management – *Physical Design*; H.2.8 [**Information Systems**]: Database Management – *Database Applications* .

cadastral applications, military operations, weather systems, multimedia presentations, moving objects etc. Some of the objects managed in these applications have associated the location and the shape as spatial attributes and both of these may evolve in time. Others, like the moving objects, have no need to store their extent, the only important spatial information being the position in space. It can be noticed that some of these applications require data that evolve discretely in time. For example, the cadastral applications or multimedia presentations deal with spatial objects (the shape of the land parcels, and the position and shape of multimedia objects, respectively) that are not changing continuously. Therefore, for such an application, the manner in which the timestamps are associated with spatial objects is straightforward. On the other side, there are also objects that may suffer a continuous evolution in time. The transportation systems that are monitoring moving objects (cars, ships etc.) have to deal with permanently changing positions in the working space.

The main decision that has to be taken during the modeling operation of ST data is how to put together the space and time elements. The simple addition of one dimension to the other may lead to different advantages in performing queries given to one domain (spatial or temporal) to the others detriment. The real challenge is combining space and time in a way that does not put to advantage one of them, as for the storage of data and the allowed operations and queries to be performed on represented data.

Following different approaches in perceiving ST data, modeling techniques and database models, many conceptual models have been designed and concrete applications have been implemented. Some of the models represent space and evolving spatial objects organized in time-stamped layers (see *The Snapshot Model* in [5]). One layer contains the state of a geographic distribution at a moment of time, but there are no explicit temporal connections between layers. One of the earliest data models that can represent spatial and temporal information into a unified view is the Worboy's ST data model [11]. It is an object-oriented data model, within which the main entity is the *ST-atom*. The ST-atom encloses unchanging spatial information during a certain time interval. A ST object is represented by a set of ST-atoms as a *ST-complex*. These elements can be seen as the temporal extension of the spatial simplices and simplicial complexes [3]. The discrete evolution of spatial objects (with or without shape) can be represented using this data model.

An original approach is found in [12]: the *Three-Domain Model* separates semantic domain from spatial and temporal domains. The advantage of this model arises from the independence of the three domains at semantic and behavioral level. There are links from semantic and temporal objects to spatial objects and from spatial and temporal objects to semantic objects. Assuming that a spatial object is located in time, there are no direct links from semantic to spatial domain. The particular case of objects without temporal measures is marked with a null

time value. The ST data is organized within four relations: three relations that correspond to the three domains and a relation that links the semantic objects, the time elements and the spatial entities. These structures store the discrete evolution of a region's partitions, as the land usage within a certain area. Therefore, the spatial domain is the same, and only the partition changes over time. These changes (the splitting and the aggregation of land parcels) are maintained using a spatial graph. The space table records only the current spatial elements, because the older spatial objects are determined using the transitions modeled within the spatial graph.

A parametric ST model called *Parametric k-Spaghetti* is introduced in [2]. The evolving spatial data can be of type point, line segment or region. One geometry element is represented by one or more triangles (degenerate in the case of points and line segments). Therefore the ST information is stored within tuples which contain the object's id, the parametric coordinates of one triangle and a valid time interval as timestamp. Though the structure of the relation is relatively simple, the represented information can capture the continuous evolution of spatial objects in time.

Moving Object Data Models have been developed to deal explicitly with continuously moving objects. The Moving Objects Spatio-Temporal data model (MOST) [8, 10] introduces the notion of dynamic attribute represented as functions of time in order to denote an attribute that change continuously. The considered dynamic attributes are the spatial coordinates of the position of moving objects; therefore the model can represent the continuous evolution of spatial objects of type point. Each dynamic attribute A is composed by three sub-attributes: $A.value$, $A.updatetime$, and $A.function$, where $A.value$ represents the value of the corresponding dynamic attribute at time $A.updatetime$ and $A.function$ gives the evolution of the attribute's value until the next update time.

The conceptual ST data modeling process proposed in [7] materialized into a data model called 3SST. Using this model, the designer is allowed to include ST objects, but also thematic objects, without any spatial or temporal attributes. Depending on the application, events may be modeled using particular event-type objects. Discrete and continuous evolutions are allowed for the spatial objects which can be points, lines or simple polygons (therefore, spatial objects with or without shape).

In this paper the work on the 3SST conceptual model is continued by proposing a concrete model in order to be implemented on a relational database system.

The paper is organized as follows: the next section presents the characteristics and the diagram corresponding to the final step of the 3SST conceptual data modeling process. The concrete relational 3SST model is introduced in Section 3 and Section 4 presents a classification of spatial and temporal operations. The final section contains conclusions and proposed future work.

2. The 3SST Conceptual Data Model

A weakness of many existing models is that each of them deals with some common characteristics found within specific applications. The paper [7] proposed a modeling process of ST data in three steps: the construction of an entity-relationship ST model, the specification of the domain model and the design of a class diagram which includes the objects characteristic to a ST application and other needed elements. The modeling phases do not take into account a certain ST application, but tries to identify and use the objects and elements needed within an application dealing, among others, with ST data. The modeling process was called 3SST (*Three Steps Spatio-Temporal* modeling process) and materialized into a data model called 3SST as well.

During the construction of the diagrams corresponding to the three modeling steps, the concrete example of a meteorological application was considered. The main entity of such an application is the meteorological phenomenon. This is one of the most complex ST objects because it is a spatial object with both position and extent as *spatial attributes*, both of them usually evolving in time. Besides these attributes, the so-called *thematic characteristics* (non-spatial attributes) of objects are considered. For example, a meteorological phenomenon may have associated: a type (rain, drizzle, fog, snow, hail, glazed frost, storm), which is a *non-temporal attribute*; different meteorological parameters (atmospheric pressure, air temperature, soil moisture, visibility, wind speed), which can be *temporal attributes* if their evolution in time is recorded. The former kind of attribute will also be called *static attribute* (an attribute whose values does not evolve in time), and the later - *dynamic attribute* (its values are changing in time and we are interested in keeping knowledge about the temporal trajectory of these values).

In order to generalize the domain of the problem and to achieve a fairly comprehensive model, the set of object types to be considered was enlarged. For example, the domain of a meteorological application may contain spatial objects with no temporal characteristics (like table-land or town), temporal objects without spatial attributes (for example the usage of equipments or the measurement of the values corresponding to different parameters) or objects with no spatial or temporal attributes (for example the persons who are studying and analyzing the meteorological information).

Another aspect taken into account was the possibility of having knowledge about the context in which the state of an object changed. If a database contains only ST objects (the attributes and the evolution of their values in time), the types of queries that may be efficiently answered are object-oriented, spatial-oriented, time-oriented and combinations of these. Yet, it cannot be known what caused the change of the state. Therefore, am other object considered was the event in order to incorporate this information [6]. An event object has usually associated as attributes the time instant and the position where the event occurred. It is

important to notice that the event objects have no evolution in time and that an event is not a ST object, even if it has associated spatial and time elements.

An observation is made in order to clarify the difference made by this paper between time and temporal elements. It is called *temporal* an element (object or attribute) whose state (value) is changing over time. The timestamps associated with evolving elements are simply called *time* elements.

After the first two modeling steps (the construction of the E-R and the 4-Domain diagrams), the presented data model was able to represent four types of objects:

- *Non-spatial non-temporal objects* (noted here as *thematic objects*): the objects that do not have any spatial or temporal attribute;
- *(strict) Spatial objects*: they have at least one spatial attribute, but its evolution in time is not recorded, and do not have any temporal attribute;
- *(strict) Temporal objects*: these objects do not have any spatial attribute, but they have associated at least one valid time or transaction time attribute;
- *Spatio-temporal objects*: they have at least one spatial attribute whose evolution in time is recorded.

The class diagram corresponding to the 3SST data model is depicted in figure 1. This is the result of the normalization operations over the object classes [1], and only the class model in 3ONF (the 3rd Object Normal Form) is presented. The normalization process at object class level was used during the refining operations because it has the main advantage over the relation normalization the fact that it makes possible to identify independent objects not only at characteristics level, but at behavioral level as well. In this way, the obtained model is closer to a concrete one. For example, the space and time objects might be treated in a similar fashion regarding the resemblance between the spatial and temporal dimensions characteristics; nevertheless, the two domains present major differences at the behavioral level.

Some observations have to be made regarding the class diagram presented in figure 1:

- A geometric object is represented by one or more n-dimensional points (in the conceptual view). Thus, such an object can represent a point, a line segment (if there are associated two points) or any region implemented as a polygon having at least three vertices.
- The points of a polygon are stored in counterclockwise order, in order to facilitate the implementation of different computations, like area, direction, intersection, or triangulation of regions; the attribute Next_point is a link to the next point within the current list of points (if it is not the last point of the list).

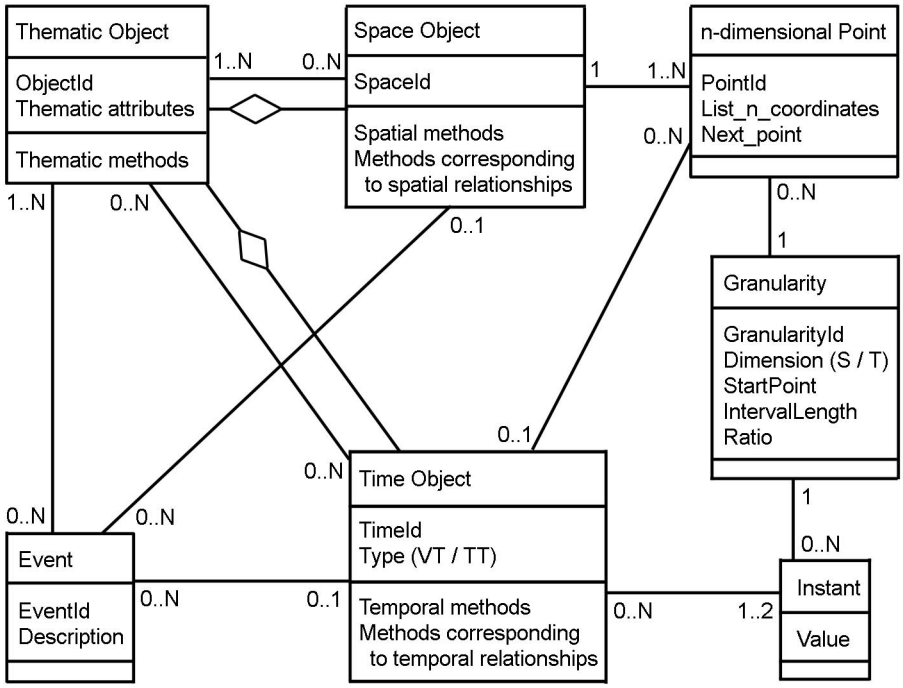


FIGURE 1. The class diagram of the 3SST conceptual data model.

- The presented spatio-temporal model allows both types of time objects to be used: the *valid time* (the time when the fact is true in the modeled reality) and the *transaction time* (the time when a fact is stored in database).
- The time elements can be instants or intervals.
- Two different relations between thematic and space objects, and between thematic and time objects are depicted in figure 1: the case of a thematic object having a time attribute or a spatial attribute that does not evolve in time implies the relation of association; if the object has thematic or spatial dynamic attributes, then the composition relation is considered.
- The entity *Granularity* is included in order to express spatial, time and numerical data in association with different measurement units.
- The methods that define the behavior of thematic objects, space objects and time objects are not presented in detail: the methods of thematic objects may be implemented according to the nature of managed data, and the methods for spatial and temporal data correspond to different

spatial and temporal operators. A classification of spatial and temporal operators is given in Section 4.

- The values of a spatio-temporal attribute may evolve discretely or continuously. On the other hand, regarding the spatial attributes of objects, the changes that may occur are on shape and / or position. These kinds of changes are represented by the data members of the *Point* class: if a point object is represented by a scalar value or a constant function of time, the evolution of that point is discrete on a corresponding time interval or it is a static object; the continuous evolution of a point object during a time interval might be represented by a non-constant function of time.

3. The Relational 3SST Data Model

Some aspects that had been considered during the design and implementation operations of the relational structures are mentioned next:

- The embedding space is considered to be two-dimensional during the implementation phase, because of some limitations of the used Transact SQL language and the implementation of some spatial operators.
- The time elements are enriched with an attribute that represents the corresponding time zone, if needed. For example, the timetable of airplanes uses the local time for arrivals, but, in order to be able to compute the duration of a flight, the difference between the time zones is needed to be known.
- It is known that most of the spatial databases that use the vector data model represent spatial entities by their approximations: a line is represented by as set of connected line segments and a region is modeled as polygon. The implementation of the 3SST model makes use of the same technique. The set of polygons that are represented by a set of points (the vertices) are convex or non-convex and have to be simple, non-self-intersecting polygons. The next sub-section contains the definitions of the basic spatial data types used within the 3SST model: point, line segment, line, and polygon.

3.1. **The Represented Spatial Data.** The space Sp that includes the spatial objects is considered theoretically to be the Euclidian n-dimensional space. Because of implementation reasons, the domain of values corresponding to the coordinates of points is limited to the *Real* type of the system.

According to the above observation, $Sp = R^2 = \{(x_1, x_2) \mid x_1, x_2 \in R\}$. Let $P = (x_1,\ x_2)$, $x_i \in R$, $i := 1..2$ be a *point* of the considered space.

Let $P_1,\ P_2 \in Sp, P_1 \neq P_2$ be two points. The *line segment* defined by P_1 and P_2 is given by $S = \{P_s \mid P_s = a \times P_1 + (1 - a) \times P_2, a \in [0, 1]\}$.

In order to define the line and polygon entities, the *oriented line segment* is

considered to be the vector determined by two points P_1 and P_2. Therefore, if $P_1,\ P_2 \in Sp,\ P_1 \neq P_2$, and $SO_1 = (P_1,\ P_2)$ and $SO_2 = (P_2,\ P_1)$ are two oriented segments, then $SO_1 \neq SO_2$

The *line* is defined as a set of oriented segments, $L = (SO_1,\ SO_2,\ ...,\ SO_l)$, such as:

PL1: $\forall i := 1..(l-1),\ SO_i.P_2 = SO_{i+1}.P_1$ (the segments are connected at their end points);

PL2: $\forall i, j := 1..l,\ i \neq j,\ SO_i \cap SO_j = \oslash \vee SO_i \cap SO_j = \{P\}$ (the segments are not overlapping, partially or totally).

Let $Pg = (SO_1,\ SO_2,\ ...,\ SO_p)$, $p \geq 1$, be a set of oriented segments. Pg is a *simple polygon* if the following conditions are fulfilled:

PP1: $\forall i := 1..p,\ SO_i.P_2 = SO_{(i+1)MODp}.P_1$ (the segments are connected at their end points);

PP2: $\forall i := 1..(l-2),\ j := (i+2)..l,\ SO_i \cap SO_j = \oslash$ (any two non-consecutive segments are not intersecting);

PP3:

$$\sum_{i:=2}^{p-1} A(\triangle P_1 P_i P_{i+1}) > 0 \tag{1}$$

($A(\triangle P_1 P_i P_{i+1})$ is the signed area of the triangle $\triangle P_1 P_i P_{i+1}$, i:=2..(p-1); the sum of the triangles signed areas represent the signed area of the polygon; the positive sign assures the counterclockwise orientation of the vertices of the polygon).

3.2. **The Time Elements.** The type of time objects that can be used as associated timestamps to thematic or spatial values is instant or interval. It is considered that the time domain is the time of reality, and not simply a surrogate temporal axis, as the real numbers.

The evolution of an object O is given by a sequence of states $(S^1,\ S^2,\ ...,\ S^n)$, each state being defined over a certain time interval (in the case of evolutions represented on time intervals). Let I^k with the end points t_1^k and t_2^k, $t_1^k < t_2^k$, be the time interval corresponding to S^k state, k:=1..n. If the lifespan of O is continuous and there cannot exists two different states of O at the same time, then any two time intervals I^k and I^j, $k, j := 1..n,\ k \neq j$, must be disjoint and they are closed at their "left" end point and open at the "right" end point. Figure 2 depicts a discrete evolution of O, which consists of four states $(S^1,\ S^2,\ S^3,\ S^4)$, during time interval $[t^1,\ t^5)$.

3.3. **The Relational Implementation of the 3SST Model.** The structure of the 3SST relational model described in this section (see figure 3) corresponds to the presented conceptual ST model. The property of generality invoked during the conceptual modeling operation is inherited by the relational model. Therefore, the 3SST relational model can be considered the core of a ST data model. For example, the current structure contains one relation *Object* which includes the

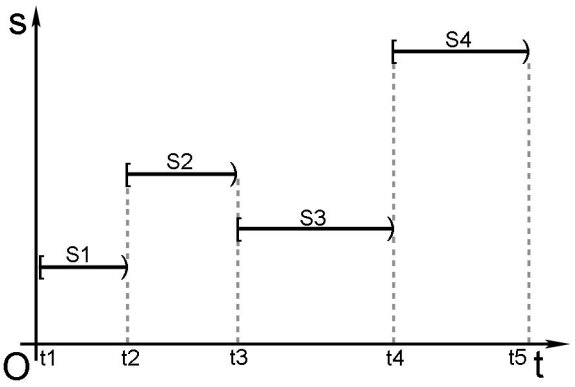


FIGURE 2. The four states of an object's discrete evolution over the time interval [t1, t5).

static data of the application domains entities. Depending on the set of objects, more *Object*-like relations can be included in the database.

A set of observations and comments about the diagram structure depicted in figure 3 are given next:

- A measurement unit is included into a single family of granularities, each of these having a parent granularity.
- Each spatial element has associated a unique ID (SOID) and each point identified by PID corresponds to a certain spatial element (see the foreign key *Point* (SID) referencing *Spatial_Obj* (SOID)).
- The type of the ROW_ID columns is *Timestamp*. The chosen data type assures that the contained values are unique within the entire database, not only within a relation. Another advantage is that their values are automatically managed by the server.
- A static point does not have associated any tuple within the relation *Evolution*.
- Any point is identified by the value PID, but a state of a point is identified by ROW_ID. Therefore, the evolution of a point is given by the set of tuples of the *Point* relation that contain the given PID value.
- The relation *Evolution* contains the complete evolution of all application's objects (non-spatial and spatial).

4. Operations on Spatio-Temporal Data

This section presents a classification of spatial and temporal operations that can be usually performed on ST data, and examples of operations are given for each category of operations. The implementation of these operations is already

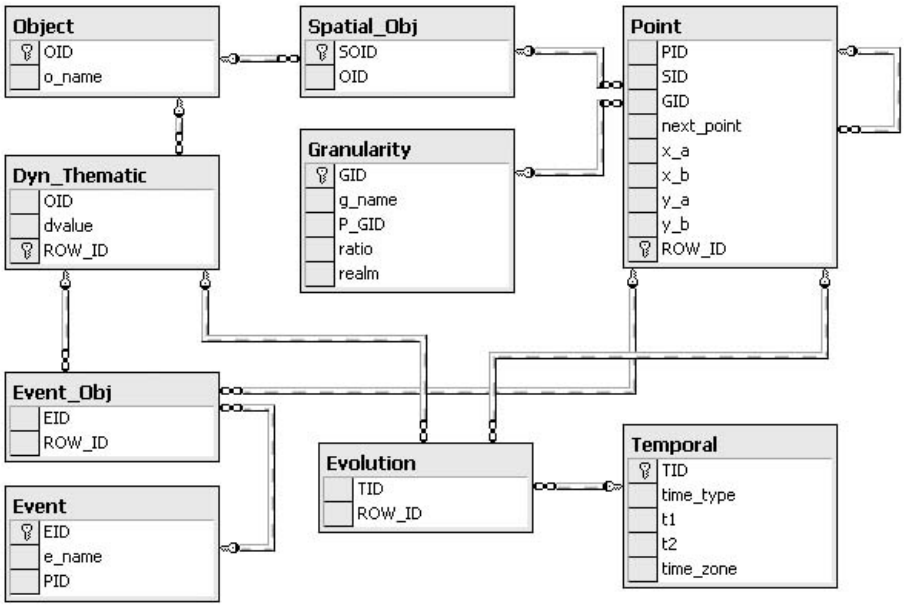


FIGURE 3. The diagram of the 3SST relational model.

accomplished or it is in progress (the aggregation operations).

The spatial and temporal operations on ST data are:

- Operations with numerical result:
  - Simple:
    - Spatial (the distance between two points, the perimeter of a polygon, the area of a triangle etc.);
    - Temporal (the length of a time interval etc.);
  - Aggregation:
    - Spatial (the total area covered by the evolution of an object of type region);
    - Temporal (the average length of a set of time intervals);
- Operations with Boolean result (predicates):
  - Topologic:
    - Spatial (the intersection of two polygons, the inclusion of a point into a polygon)
    - Temporal (a time instant is included into a time interval, the adjacency of two time intervals, the intersection of two time intervals)
  - Metric:

        * Spatial (in_circle, in_square)
        * Temporal (in_neighbourhood)
    - Directional
        * Spatial: the current implementation includes eight functions with result of type Boolean, corresponding to the directional operations mentioned next (is_north, is_south_east etc.)
        * Temporal (an instant of time is before a time interval)
- Operations with result of type Direction - only for spatial data of type point; there are eight values used of type Direction E, NE, N, NW, W, SW, S, SE. Given two points, P_1 and P_2, the result is returned according to the angle θ, where θ is determined by the two semi-lines $[P_1P_1'$ and $[P_1P_2$, $P_1'.x = P_1.x + 1$, $P_1'.y = P_1.y$;
- Operations with result of type spatial or time:
    - Selection of an object's component (the geometric state of a given object at a certain time instant);
    - Simple, construction (the intersection, reunion, difference - spatial or temporal - between two spatial or temporal elements);
    - Aggregation, construction:
        * Spatial (the trajectory of a mobile object as the projection in the 2D space of its spatial evolution)
        * Temporal (the lifespan of a given object)

These operations are used within the implementation of what is considered to be ST operations; therefore, the later ones will not be explicitly mentioned. For example, a ST window query like "Which are the objects that passes through region R during time interval T?" is solved by querying objects whose 3D trajectory intersects the 3D region defined by R X T. Therefore, the mentioned query is reduced to an intersection query of the valid trajectory during T with the region R.

The output of an operation with a complex result is represented as a recordset (for example: a set of points representing the state of a geometric object at a given time instance).

According to the author's knowledge, this is the first paper that introduces the data type *Direction* and mentions operations that return a value of this type.

The operation routines and queries are currently written using the MS SQL-Server's Transact-SQL language.

5. Conclusions and Future Work

The presented paper shows the capability of a relational database system to store ST data with discrete or continuous evolution in time. The spatial attributes of considered objects may be of type point, line or simple polygon. The implementation of spatial evolution allows the defining points of the geometric objects

to change independently, with a different frequency, on different time intervals. Also, the implemented model is able to perform different spatial, temporal and ST operations and queries on the stored data, with the help of a set or routines written in the standard query language.

The current work will be continued by the implementation of aggregation operators and visual interface. In order to implement in a more elegant and efficient fashion the data and the corresponding routines presented in the conceptual model, the proposed future work also includes the implementation on top of an object-relational database system and the study of queries performance on a large set of data.

References

[1] S. W. Ambler, *Introduction to Class Normalization*, Available at www.agiledata.org, last updated 2006.

[2] J. Chomicki, P. Revesz, *Constraint-Based Interoperability of Spatiotemporal Databases*, Proc. of the 5th Intl. Symposium on Large Spatial Databases, Springer-Verlag, LNCS 1262, 1997, pp. 142-162.

[3] M. Egenfofer, A. Frank, J. P. Jackson, A Topological Data Model for Spatial Databases, Proc. of the 1st Intl. Symposium on Large Spatial Databases, Santa Barbara, 1989, pp. 271-286.

[4] C. S. Jensen, R. T. Snodgrass, Temporal Data Management, TimeCenter TR-17, 1997.

[5] G. Langran, N. R. Chrisman, A Framework for Temporal Geographic Information Systems, Cartographica, 25, 3, 1988, pp. 1-14.

[6] D. Peuquet, N. Duan, An Event-Based Spatio-temporal Data Model (ESTDM) for Temporal Analysis of Geographical Data, Intl. Journal of Geographical Information Systems, 9, 1, 1995, pp. 7-24.

[7] A. Sabau, The 3SST Model: A three Step Spatio-Temporal Conceptual Model, Proc. of the 2nd AIS SIGSAND European Symposium on System Analysis and Design, Gdansk, 2007, pp. ?-?.

[8] A. P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Modeling and Querying Moving Objects, Proc. of the 13th Int. Conf. on Data Engineering (ICDE13), 1997, Birmingham, 422-432.

[9] R. T. Snodgrass, Of Duplicates and Septuplets, Database Programming and Design, 11(6), 46-49, 1998.

[10] O. Wolfson, B. Xu, B., S. Chamberlain, L. Jiang, L., Moving Objects Databases: Issues and Solutions, Proc. of the 10th Intl. Conference on Scientific and Statistical Database Management (SSDBM98), 1998, pp. 111-122.

[11] M. F. Worboys, A Unified Model for Spatial and Temporal Information, The Computer Journal, 37, 1, 1994, pp. 27-34.

[12] M. Yuan, Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries, Transactions in GIS, 3(2) , 1999, pp. 137-159.

Faculty of Mathematics and Computer Science, Babes-Bolyai University, Cluj-Napoca
E-mail address: deiush@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

THE APPLICABILITY OF FUZZY THEORY IN REMOTE SENSING IMAGE CLASSIFICATION

GABRIELA DROJ

Abstract. In the recent years, the usage of Geographic Information Systems has been rapidly increasing and it became the main tool for analyzing spatial data in unprecedented number of fields of activities. The evolution of GIS led to the necessity of faster and better results. The processing time was reduced by using more and more advanced and applied mathematical and computer science knowledge. One of these mathematical theories is fuzzy logic. The fuzzy logic theory gives the possibility of enhancing spatial data management with the modeling of uncertainty. The usage of fuzzy theory has also applicability in processing remote sensing data. In this paper is presented the applicability of fuzzy set theory to the classification of raster images

1. Introduction

Geographical Information System (shortly GIS) represents a powerful set of tools for collecting, storing, retrieving at will, transforming and displaying spatial data from the real world. GIS extends the limits of Computer Aided Design (CAD) and Automated Mapping (AM) with the possibility of retrieving geospatial data at request and with the possibility of "what if" analysis and scenarios.

The difference between a CAD system and a GIS system is given by the possibility of spatial analysis, where new maps are computed from existing ones by applying either:

- Spatial Join operation between different geospatial data;
- Spatial aggregation;
- Buffer operations, which increase the size of an object by extending its boundary;
- Set operations, such as complement, union, and intersection.

Received by the editors: July 2007 .

2000 *Mathematics Subject Classification.* 68U10, 03E72, 03E75.

1998 *CR Categories and Descriptors.* I.4.6. [**Computing Methodologies**]: Image Processing and Computer Vision – *Segmentation*; I.5.3.[**Computing Methodologies**]: Pattern Recognition – *Clustering* F.4.1.[**Theory of Computation**]: Mathematical Logic and Formal Languages – *Mathematical Logic* .

In traditional GIS, these operations are exact quantitative operations. Humans, however, often prefer a qualitative operation over an exact quantitative one, which can be achieved by extending the standard map, overlaying operations to fuzzy maps and using fuzzy logic rather than crisp logic.

Fuzzy logic was implemented successfully in various GIS processes like:

- Data collection – analysis and processing remote sensing data for classification algorithms and object recognition;
- Spatial analysis - processing qualitative data, defining relationships between uncertain geospatial objects;
- Complex operations based on genetic algorithms or artificial intelligence; in this category we can include also the object recognition from airborne images.

Remote sensing from airborne and space borne platforms provides valuable data for mapping, environmental monitoring, disaster management and civil and military intelligence. However, to explore the full value of these data, the appropriate information has to be extracted and presented in standard format to import it into geo-information systems and thus allow efficient decision processes. The process usually used for extracting information from these images is based on classification.

Classification or clustering is the process of automatically grouping a given set of data into separate clusters such that data points with similar characteristics will belong to the same cluster. In this way, the number of clusters is reduced. The process of classification is usually based on object's attributes or characteristics than on its geometry but in the process of image classification is based on multi spectral analysis of the pixels. An image classification is acceptable if the distortion of the image is minim.

This paper will present the applicability of fuzzy set theory in classification of raster images. In the same time is presented the possibility to optimize the classification procedure by using fuzzy set theory to obtain a minimum distortion of the image.

2. On Classification Methods

In classical cluster analysis each pixel must be assigned to exactly one cluster. Fuzzy cluster analysis relaxes this requirement by allowing gradual memberships, thus offering the opportunity to deal with data that belong to more than one cluster at the same time.

The result of clustering using fuzzy classification consists of a multi-layer output file, one layer for each cluster. Each layer can be saved as an independent image. In the image layer, the black is representing the membership 0 and white is representing the membership 1. The pixels in different gray tones are representing the degree of membership to the cluster.

The theory of fuzzy sets has its main applicability in the process of raster classification

2.1. **Unsupervised Fuzzy Classification.** The unsupervised classification is a completely automatic process; it's eliminating the user and in the same time the influence of known information. This kind of clustering procedure is used where are no information about the site represented in the image to be classified.[Tur]

The most popular unsupervised classification methods are ISODATA and the k-means algorithm. [Eas01, Su03] There are many unsupervised classification algorithms based on fuzzy set theory, we mention fuzzy c-means, fuzzy Gustafson – Kessel algorithm, fuzzy c - shells and genetic algorithm and so on.[Ben03, Tur]. One of the most popular fuzzy based unsupervised classifications is the Fuzzy c-means algorithm, similar to K-means algorithm.

2.2. **Supervised Classification.** The results of classification can be optimized if there are known information about the image to be clustered. In this case the method used is called supervised classification. The supervised classification is made in two steps. The first step is to create signature files (training sites) and the second step is the classification itself. The fuzzy logic can be used in the creation of the signatures files and also in the process of the classification itself, but is not necessary to be used in both stages. [Ben03, Eas01, Liu03, Ste99, Su03]

The signatures files based on this training site can be created using unsupervised classification methods (like ISODATA), classical methods (like Maximum Likehood or Minim Distance) or by computing a fuzzy matrix filled with the values indicated by the membership grade of each training site [Eas01]. The membership degree is computed by using fuzzy membership functions like: piecewise linear functions, the Gaussian distribution function, the sigmoid curve and quadratic or cubic polynomial curve. The evaluation on the training can be done using statistical methods: minimum, maximum, mean, and standard deviation for each band independent and covariance matrix for all the three bands. The most relevant signature file evaluation is creating an error matrix as a matrix of percentages based on pixel counts that allows us to see how many pixels in each training sample were assigned to each class.

The second step of the supervised classification can also be processed with traditional methods (Minimum distance, Mahalanobis distance or Maximum Likehood) or by using fuzzy membership functions or genetic algorithm. This stage cannot be evaluated alone, it can be evaluated just the final result of the supervised classification.

In order to optimize the accuracy of the results we consider that is appropriate to use fuzzy logic in both stages of supervised classification. In the next chapter we are testing these hypotheses.

2.3. **Fuzzy convolution.** Independently of the classification method used, in order to optimize the results we propose usage of the fuzzy convolution operation. Fuzzy convolution creates a single classification band by calculating the total weighted inverse distance of all the classes in a window of pixels and assigning

the center pixel the class with the largest total inverse distance summed over the entire set of fuzzy classification bands. This has the effect of creating a context-based classification to reduce the noise of the classification. Classes with a very small distance value remain unchanged while classes with higher distance values may change to a neighboring value if there are sufficient number of neighboring pixels with class values and small corresponding distance values.

3. Experiments with real-world data

3.1. **Input data.** For the procedures of image classification was used an orthorectified airborne image from the upper hills of Oradea municipality. This image contains three channels recorded in three bands: the first band for green, the second for red and the third for blue. In the figure below, we present a fragment of this image and some statistics for the whole image.

Figure 1. Image fragment and statistics

Statistics	Band1	Band 2	Band 3
Min value per cell	9	8	9
Max value per cell	255	255	227
Mean Deviation	127	138	109
Mean	127	137	108
Standard deviation	31.330	22.637	17.801
Correlation	2.7	0.169	0.061

3.2. **Definition and verification of the training areas.** Training is the first stage of a supervised classification. In this step the user must define training areas for each class interactively on the displayed image. The areas may be specified both on polygon and on pixel basis. The three classes of information defined are:

streets, houses and green areas (figure below). The signature files were created using a fuzzy membership function based on the sigmoid curve, in this case the accuracy of the signature files are acceptable, over 90 %(table below).

FIGURE 2. Training sites

Data	Streets	Houses	Green Areas
Streets	95.45	4.87	1.27
Houses	2.73	93.42	2.83
Green Areas	1.82	1.71	95.89

3.3. **Classification procedure.** In order to obtain a minimum distortion of the classified image it was studied the possibility of optimizing the classification procedure by using fuzzy theory. We analyzed the result of classification procedures in three experiments.

In the first experiment, the classification procedure is done using unsupervised classification, by using c-means algorithm. The result of fuzzy c-means algorithm on the input data, for three clusters is represented in the set of images presented in figure 3.

In order to optimize the accuracy of the results we consider that is appropriate to use fuzzy logic in both stages of supervised classification. In the following test cases, we are testing these hypotheses. In the second case we studied the result of classification by using fuzzy set theory just for making the signature file and in the third case we analyzed the results of using fuzzy theory in the both stage of supervised classification.

The actual classification process can be done by using fuzzy membership function or based on the standard distance of each pixel to the mean reflectance on each band for a signature.

On the second experiment we used a method based on the standard distance of each pixel till the relevant pixel. The result of the clustering on the chosen image,

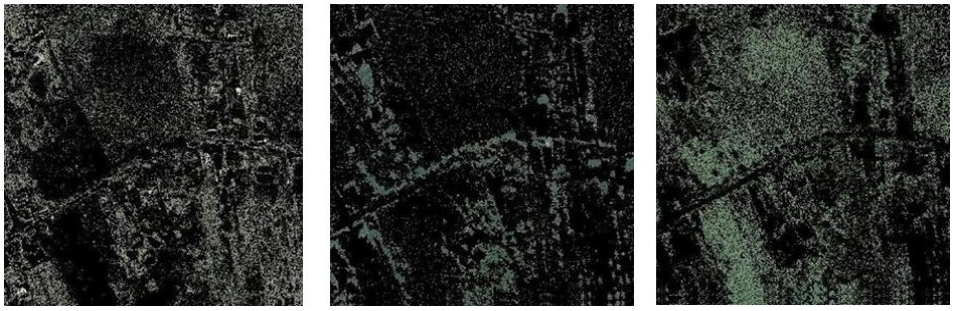


FIGURE 3. Unsupervised Fuzzy Classification - C-means

with the signature file defined before, is represented in figure 4, in three images, one for each cluster:

- The image in the left for the class representing the houses,
- The image in the middle for the class representing the green areas,
- The image in the right for the class representing the streets

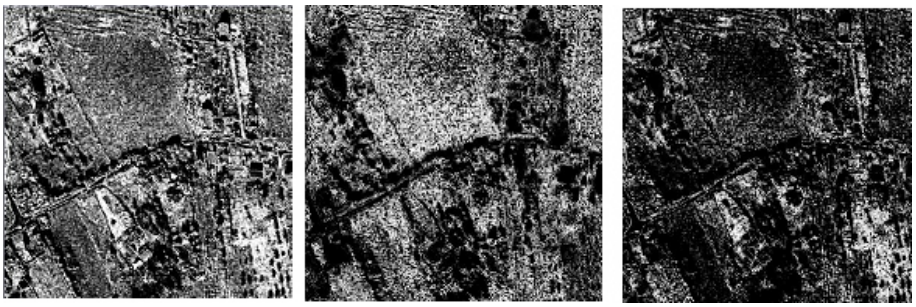


FIGURE 4. Supervised Fuzzy Classification - First Case

For the third experiment of classifying the input images, with the signature files defined before, we used a classifying method which combined the Maximum Likehood method with a fuzzy membership function (linear). The output images are represented in the figure 5, each image represent a cluster in the following order houses, grean areas, streets.

3.4. **Fuzzy convolution.** Classification results can be optimized through fuzzy convolution. The scope of this procedure is to combine the previously created bands in one band by interpolation. The interpolation method used is the weight inverse distance. Applying the fuzzy convolution on the results obtained after the

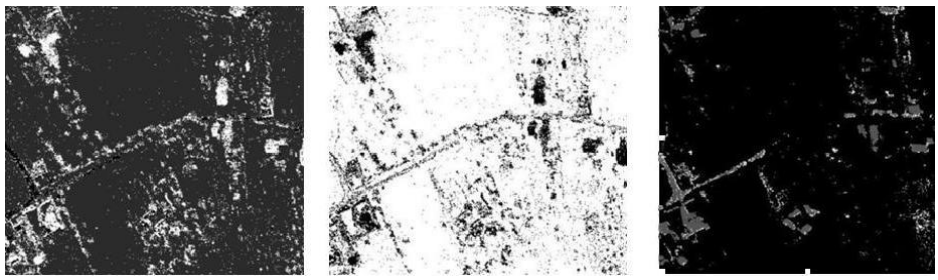


FIGURE 5. Supervised Fuzzy Classification - Second Case

second case of supervised classification, the image represented in Figure 6 was obtained.

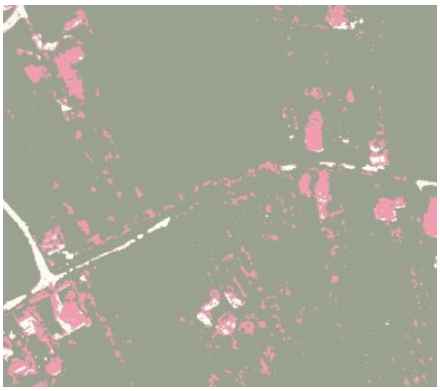


FIGURE 6. Fuzzy Convolution

4. RESULTS OF EXPERIMENTS

The results of classification based on fuzzy method were presented in the previous section, within the figures 1, 2 and 3. As the images show, the supervised classifications have higher quality than the unsupervised classification. For the evaluation of the fuzzy classifications presented before, three sets of accuracy measures have been considered:

- Overall accuracy – based on percent of correct identification;
- Overall kappa coefficient;
- Overall correlation coefficient

	Unsupervised	Supervised - first case	Supervised - second case
Overall accuracy	34%	72%	83,9%
Kappa	0.1151	0.3421	0.4476
Overall Agreement	0.31	0.69	0.89

5. Conclusion

The major advantage of Fuzzy logic theory is that it allows the natural description of data and problems, which should be solved, in terms of relationships between precise numerical values. Fuzzy sets make possible not only the definition of uncertain, vague or probabilistic spatial data, but also allow relationships and operations on them.

The usage of fuzzy theory has implications in improving the quality of the classification of airborne images and object recognition. The Fuzzy set theory offers instruments for supervised and unsupervised classification. The unsupervised fuzzy based classification allows clustering of data, where no a priori information known (c-means algorithms), but the supervised classification is offering higher quality.

The Fuzzy set theory represents a powerful instrument in designing efficient tools to process remote sensing images and also to support the spatial decision-making process.

References

[Ben03] Benz, U.; Hofmann, P.; Willhauck, G.; Lingenfelder, I.; Heynen, M. - Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information, Journal of Photogrammetry & Remote Sensing, 2004

[Eas01] Eastman, R – Idrisi 32 Release 2, Guide to GIS and Image Processing, Volume I- II, Clark University, May 2001

[Gue03] Guesgen, H.;Hertzberg, J.; Lobb, R.;Mantler, A. – Buffering Fuzzy Maps in GIS, Spatial Cognition and Computation, 2003

[Liu03] Liu, H., Li. J., Chapman, M. A.;- Automated Road Extraction from Satellite Imagery Using Hybrid Genetic Algorithms and Cluster Analysis, Journal of Environmental Informatics 1 (2) 40-47 (2003)

[Ste99] Stefanakis, E.; Sellis, T.- Enhancing a Database Management System for GIS with Fuzzy Set Methodologies, Proceedings of the 19th International Cartographic Conference, Ottawa, Canada, August 1999.

[Su03] Su-Yin Tan - A Comparison of the Classification of Vegetation Characteristics by Spectral Mixture Analysis and Standard Classifiers on Remotely Sensed Imagery within the Siberia Region, International Institute for Applied Systems Analysis Austria, 2003

[Tur] Turčan, A.; Ocelíková, E.; Madarász, L. – Fuzzy c-means Algorithms in Remote Sensing, Technical University of Kosice

City Hall of Oradea, GIS Department

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

ON INDEPENDENT SETS OF VERTICES IN GRAPHS

V. CIOBAN

Abstract. Among the remarkable sets of vertices of a graph, the independent sets of vertices (acronym **IS**, other name is the internal stabile sets of vertices) are important, because using them we can solve many classification and counting problems: in chemistry, automobiles traffic, espionage, the chess game, the timetable problem. A new kind of independent sets of vertices fuzzy independent sets, related to fuzzy graphs are defined, and an algorithm for finding fuzzy independent sets (**FIS**) is given in this paper.

1. Introduction

There are several algorithms that determine the maximal **IS**-s of a graph. In [4] and [8], the authors have proposed an algorithm based on the calculus of associated boolean expressions of a graph. Another algorithm is due to Bednarek and Taulbee (which may be seen in [7]). A recursive version of this algorithm was given in [1]. Also, a simplified version of this algorithm that works with only one list was given there.

An algorithm based on the maximal complete submatrices was proposed by Y.Malgrange [5]. Using the Malgrange's relations a new and more clearly algorithm was given in [3].

These algorithms have an algebraic or a combinatorial character, and have $O(n^2)$ or $O(n^3)$ complexity.

2. Fuzzy independent sets of vertices in a fuzzy graph

Definition 2.1. A fuzzy graph is a graph $G = (V, \Gamma)$ with labeled edges. Every label is a number from $(0, 1]$ and shows the degree of the vertices association.

For a given fuzzy graph $G = (V, \Gamma)$ we can associate a fuzzy matrix (the degree of the association of every related vertex). See, for example, figure 1.

The fuzzy matrix is:

$$M = \begin{pmatrix} 0 & 0.2 & 0 & 0 & 0 \\ 0.2 & 0 & 0.9 & 0 & 0 \\ 0 & 0.9 & 0 & 1 & 0.4 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 \end{pmatrix}$$

Received by the editors: July 1, 2007.

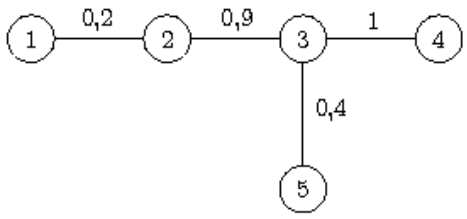


FIGURE 1

We can define the associated fuzzy matrix for a given graph:

$$M = (m_{ij}), \quad i = \overline{1,n},\ j = \overline{1,n},\ \text{with } m_{ij} = \begin{cases} \delta_{ij}, & [v_i, v_j] \in \Gamma \\ 0, & \text{otherwise} \end{cases}, \quad \delta_{ij} \in (0,1]$$

Definition 2.2. Let $G = (V, \Gamma)$ be a fuzzy graph. $S^\delta \subseteq V$ (δ is given too) is called a fuzzy independent set (**FIS**) of vertices iff $m_{ij} \leq \delta$ for each edge $[v_i, v_j]$, $v_i, v_j \in S^\delta$.

Remarks.

(1) We may define $G = (V, E)$ where E is the set of edges, $E \subseteq V \times V$; an edge is $[x, y]$, where $x, y \in V$ and we presume that E doesn't contain edges of the form $[x, x]$.
(2) Let S^δ be an **FIS**. We call S^δ maximal in relation to set inclusion, if S^δ is not included in any other **FIS**.
(3) Let S^δ_G be the family of maximal **FIS**-s of G. One can define:
   - $\alpha^\delta(G)$, the fuzzy internal stability number of the graph G as $\alpha^\delta(G) = \max |S^\delta|$, $S^\delta \in S^\delta_G$
   - $\gamma^\delta(G)$ is the fuzzy chromatic number of the graph G. If $S^\delta_1, \ldots, S^\delta_p$ are **FIS**-s with the properties: $S^\delta_i \cap S^\delta_j \neq \emptyset$, for $i \neq j$ and $\bigcup_{i=1}^{p} S^\delta_i = V$ than $S^\delta_1, \ldots, S^\delta_p$ form a chromatic decomposition of the graph G and $\Gamma(G) = \min(p)$. So $\Gamma^\delta(G)$ is the lowest number of disjoint **FIS**-s that covers G.
(4) $S^0_G = S_G$.

For the graph depicted in figure 1 we have:
1) for $\delta = 1$ the FIS family is reduced to the entire set of vertices $\{1, 2, 3, 4, 5\}$
2) for $\delta = 0.5$ the FIS family is $\{\{1, 3, 5\}\{1, 2, 4, 5\}\}$
3) for $\delta = 0$ the set FIS family is $\{\{1, 3\}\{1, 4, 5\}\{2, 4, 5\}\}$ = IS family

3. A simplified algorithm to find the (FIS) of vertices using Malgrange's relations

Malgrange has defined the relations $\overline{\cup}$ and $\overline{\cap}$, as follows. Let M^* be the family of maximal submatrices of M (M is a boolean matrix). On the set M^* Malgrange define the $\overline{\cup}$ and $\overline{\cap}$ operations:

Let m_1 and m_2 be matrices of M^*, $m_1 = (A_1, B_1)$; $m_2 = (A_2, B_2)$.

A_1 contains row numbers, let i be one of them; B_1 contains column numbers, let j one of them; then $M(i,j) = 1$ (see the example below) then:

$$m_1 \overline{\cup} m_2 = (A_1 \cup A_2, B_1 \cap B_2) \quad \text{and} \quad m_1 \overline{\cap} m_2 = (A_1 \cap A_2, B_1 \cup B_2)$$

Example. Let G be the undirected graph (depicted in figure 1), $G = (V, E)$ where: $V = \{1, 2, 3, 4, 5\}$ and $E = \{[1, 2]; [2, 3]; [3, 4]; [3, 5]\}$.

The adjacent matrix is:

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

and the complementary matrix

$$\overline{M} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

A row-cover of $\overline{M}$ is $RC = \{(1, 1345); (2, 245); (3, 13); (4, 1245); (5, 1245)\}$.

A column-cover of $\overline{M}$ is $CC = \{(1345, 1); (245, 2); (13, 3); (1245, 4); (1245, 5)\}$.

Also $(1, 1345)\overline{\cup}(2, 245) = (12, 45)$ and $(1, 1345)\overline{\cap}(2, 245) = (\emptyset, 12345)$.

For a given fuzzy graph $G = (X, E)$ a row-cover from the associated fuzzy matrix, depending on δ, is defined by RC:

$$RC = (i, j_1, \ldots, j_p) \in \text{with conditions } 0 < m_{i,j_1}, \ldots, m_{i,j_p} \leq \delta \qquad (*)$$

The following algorithm computes the **FIS** of G.

S1. One finds C_0 a row-cover from the associated fuzzy matrix, depending of δ; $FFIS := \emptyset$; $k := 1$; (FFIS is the FIS family)

S2. a) One finds C_k:
$\forall\ m_1, m_2 \in C_{k-1}$ (let $m_1 = (A_1, B_1)$ and $m_2 = (A_2, B_2)$)
If $A_1 \cup A_2 \neq \emptyset$ and $A_1 \cup A_2 \subseteq B_1 \cap B_2$ then $m_1 \overline{\cup} m_2 \in C_k$.

b) If $m \in C_k$ and $m = (I, I)$ then $FFIS := FFIS \cup I$ and $C_k := C_k \backslash m$;

S3. Repeat S2 for $k = 2, 3, , k_0$ until $C_{k_0} = \emptyset$.

Finally FFIS contains the FIS of G.

4. Example

An interesting problem is to predict the medical virus infection of certain locations (villages, towns). A graph $G = (V, E)$ is used to model this problem. V is the locations set and E represents the ways of virus propagation. Each arrow (v_i, v_j)

is labeled with δ_{ij} (the probability of virus propagation from v_i to v_j). At the beginning there are some infected locations. Let the vertex number 3 and vertex number 4 (see figure 1) be the first infected locations (denoted by $VF = \{3, 4\}$). We try to find out the future infected locations by with probability p (p is superior limit of the infection probability).

Problem solving steps are:

1. One finds out the FIS family for $\delta = 1 - p$. If $p = 0.7$ then $\delta = 0.3$ and the corresponding FIS family is

$$FFIS = \{\{1, 3, 4\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 4, 5\}\}.$$

From this family one chooses the FIS sets that include VF: $\{1, 3, 4\}$ and $\{2, 3, 4\}$.

2. From the previous sets one chooses the sets that include related vertices with every vertex from VF set. For our example the chosen set is $\{2, 3, 4\}$ (for the other set we can see that the vertex 1 is not related by vertex 2 and vertex 3 either. The problem is solved for the first period (an hour, a day, a week,...).

For the next period we'll chose $VF = \{2, 3, 4\}$ and we put the label 1 on the arrows between each pair from VF.

One can modify (if it is necessary) the labels of probability (between the vertices $V \setminus VF$) and on applying the previous steps (step 1 and step 2).

5. Conclusions

The above example tells us that the FIS family can solve many problems which are abstracted and modeled by the fuzzy graphs. So, we believe this kind of generalizations is useful to solve such problems.

A new kind of FIS can be obtained by modifying the RC definition given in section 3 ($*$):

$$RC = (i, j_1 ... j_p) \in \text{with conditions } \delta \leq m_{i,j_1}, \ldots, m_{i,j_p} \leq 1.$$

References

[1] Cioban Vasile, *On independent sets of vertices in graphs*, Studia Univ. Babeş-Bolyai, Mathematica, XXXVI, 3, 1991, pp. 11-16.

[2] Cioban Vasile, *Independent sets of vertices in graphs*, Generalization, Univ. Babeş-Bolyai, Preprint,2 1995, pp. 61-66.

[3] Cioban Vasile, *On independent sets of vertices in graphs. An algebraic algorithm*, Proceedings of the Symposium "Colocviul Academic Clujean de INFORMATICA", 100-104, Cluj-Napoca, 2003 June 24th.

[4] Maghout, K., *Sur la determination de nombre de stabilite et du nombre chromatique d'une graphe*, Compte Rendus de l'Academie des Sciences, Paris, 248, 1959, pp. 3522-3523.

[5] Malgrange Yves, *Recherche des sous-matrices premiere d'une matrice a coefficients binaries. Applications de certains problemes des graphs*, Deuxieme Congres de l'AFCALTI, oct., 1961, Gauthier-Villars, Paris 1962, pp. 231-242.

[6] Moon, J.W., Moser,L., *On cliques in graphs*, Israel Journal of Mathematics, vol.3 no.1, 1965, pp. 23-28.

[7] Tomescu Ioan, *Introduction in combinatory*, Ed. Tehnică, 1972, pp 191-195.

[8] Weissman,J., *Boolean Algebra, map coloring and interconnection*, Am. Math. Monthly, no. 69, 1962, pp. 608-613.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: vcioban@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

AXIOMATIZATION OF CREDULOUS REASONING IN RATIONAL DEFAULT LOGIC

MIHAIELA LUPEA

Abstract. Nonmonotonic reasoning is succesfully formalized by the class of default logics. In this paper we introduce an axiomatic system for credulous reasoning in rational default logic. Based on classical sequent calculus and anti-sequent calculus, an abstract characterization of credulous nonmonotonic default inference in this variant of default logic is presented.

Keywords: default logic, nonmonotonic reasoning, inference relation.

1. Introduction

Default logics formalize a particular type of nonmonotonic reasoning, called *default reasoning*. These logical systems have special inference rules, called *defaults* which permit drawing conclusions in the absence of complete information, using default assumptions. In default logics a set of facts is extended with new formulas, using the classical inference rules and the defaults obtaining *default extensions*. The elements of extensions are called non-monotonic theorems, or *beliefs*. The beliefs are only consistent formulas, not necessarily true and they can be later invalidated by adding new facts.

The versions (classical([13]), justified ([6]), constrained([14]), rational([11]) of the default logic, use different meanings of the default assumptions in the reasoning process.

The computational problems specific to default logics are:

Search problem: computing the extensions of a default theory, is $\sum_2^P -complete$.

Decision problems:

- Deciding whether a formula belongs to at least one extension of a default theory - *credulous* default reasoning, is $\sum_2^P -complete$.
- Deciding whether a formula belongs to all extensions of a default theory - *skeptical* default reasoning, is $\prod_2^P -complete$.

Due to their very high level of theoretical complexity, caused by the great power of the inferrential process, the above computational problems can be solved in an

Received by the editors: September 1, 2007.

2000 *Mathematics Subject Classification.* 03B79, 68T15, 68T27, 68T20.

1998 *CR Categories and Descriptors.* I.2[**Artificial Intelligence**]: Logic in artificial intelligence – *default logics, nonmonotonic reasoning, inference relation*.

efficient manner only for particular classes of default theories.

Credulous reasoning *versus* skeptical reasoning

Accepting alternative possibilities for extending a default theory characterizes the *credulous reasoning*. The commonsense reasoning is the human model of reasoning, by making default assumptions for overcoming the lack of information. This type of reasoning belongs to the credoulous perspective of the reasoning.

Skeptical reasoning is imposed in prediction problems, because the nonmonotonic consequences cannot be later modified, which means that derived formulas does not depend on the alternative assumptions made during the reasoning process. It is considered irrational to have the possibility to chose one belief or another one if they are contradictory.

The specific of the problem will decide the appropriate perspective for the nonmonotonic reasoning used to solve the problem.

Related Work

The problem of finding efficient algorithms and building automated proof systems for default logics was the most approached in the literature [1, 5, 7, 12, 15].

An important theoretical aspect in the formalization of nonmonotonic reasoning is the study of the inference relations and operations associated to different nonmonotonic formalisms. The specific properties (cumulativity, distribution, cautious monotonicity, proof by cases, absorption, cut) of inference operations for default logics are presented in the papers [2, 8, 16].

The research in the domain of the axiomatization of nonmonotonic reasoning formalized by default logics has begun with the paper [3] and continued with [4, 10]. The proposed axiomatic systems are based on sequent and anti-sequent calculi and characterize the credulous/skeptical default inference in propositional/predicate classical default logic.

In this paper we propose an abstract characterization of credulous default inference associated to rational default logic using the *credulous rational default sequent calculus*. This axiomatic system combines sequent calculus rules and anti-sequent calculus rules with reduction rules specific to the aplication of the defaults.

The paper is structured as follows. Section 2 presents theoretical aspects of *classical* and *rational* default logics. Two complementary systems, sequent calculus and anti-sequent calculus for propositional logic, are discussed in Section 3. In Section 4 we introduce an axiomatic system for credulous reasoning in rational default logic, based on the sequent calculus. Conclusions and further work are outlined in Section 5.

2. Default logics

A *default theory* ([13]) $\Delta = (D, W)$ consists of a set W of consistent formulas of first order logic (the *facts*) and a set D of *default rules*. A *default* has the form $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma}$, where: α is called *prerequisite*, $\beta_1, \ldots, \beta_m$ are called *justifications* and γ is called *consequent*.

A default $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma}$ can be applied and thus derive γ if α is believed and it is consistent to assumed $\beta_1, \ldots, \beta_m$(meaning that $\neg\beta_1, \ldots, \neg\beta_m$ are not believed).

The set of defaults used in the construction of an extension is called the *generating default set* for the considered extension.

The results from [9] show that default theories can be represented by unitary theories (all the defaults have only one justification, $d = \frac{\alpha:\beta}{\gamma}$) in such a way that extensions (classical, justified, constrained, rational) are preserved. In this paper we will use only unitary default theories based on propositional logic.

The versions (classical, justified, constrained, rational) of default logic try to provide an appropriate definition of consistency condition for the justifications of the defaults and thus to obtain many interesting and useful properties for these logical systems. These logics must coexist because each of them models a specific type of default reasoning, based on the reasoning context and the semantics of the assumptions.

Classical default logic was proposed by Reiter[13]. Due to the individual consistency checking of justifications the implicit assumptions are lost when the classical extensions are constructed.

Justified default logic was introduced by Lukaszewicz[6]. The applicability condition of default rules is strengthen and thus individual inconsistencies between consequents and justifications are detected, but inconsistencies among justifications are neglected.

Constrained default logic was developed by Schaub[14]. The consistency condition is a global one and it is based on the observation that in commonsense reasoning we assume things, we keep track of our assumptions and we verify that they do not contradict each other.

Rational default logic was introduced in [11] as a version of classical default logic, for solving the problem of handling disjunctive information. The defaults with mutually inconsistent justifications are never used together in constructing a rational default extension.

We denote by $Th(U) = \{X | U \vdash X\}$ the classical deductive closure of the set U of formulas.

The following definitions of classical and rational default extensions show the applicability conditions of defaults in these two variants of default logic.

Definition 2.1. [13] Let $\Delta = (D, W)$ be a default theory. For any set S of formulas, let $\Gamma(S)$ be the smallest set S' of formulas such that:

1. $W \subseteq S'$;
2. $Th(S') = S'$;
3. For any $\frac{\alpha:\beta}{\gamma} \in D$, if $\alpha \in S'$ and $\neg\beta \notin S$ then $\gamma \in S'$.

A set E of formulas is a *classical extension* of (D, W) if and only if $\Gamma(E) = E$.

The set $GD_{\Delta}^{E} = \left\{ \frac{\alpha:\beta}{\gamma} | \alpha \in E \text{ and } \neg\beta \notin E \right\}$ is called the *set of the generating defaults* for the classical extension E.

Definition 2.2. [15] Let $\Delta = (D, W)$ be a default theory. For any set T of formulas, let $\Psi(T)$ be the pair (S', T') of the smallest sets of formulas such that:

1. $W \subseteq S' \subseteq T'$;
2. $S' = Th(S')$ and $T' = Th(T')$;
3. For any $\frac{\alpha:\beta}{\gamma} \in D$, if $\alpha \in S'$ and $\neg\beta \notin T$ then $\gamma \in S'$ and $\beta \wedge \gamma \in T'$.

A pair (E, C) of sets of formulas is a *rational extension* of (D, W) if and only if $\Psi(C) = (E, C)$. The set E is the *actual rational extension* of the default theory and C is the *reasoning context*.

The set $GD_{\Delta}^{(E,C)} = \left\{ \frac{\alpha:\beta}{\gamma} | \alpha \in E \text{ and } \neg\beta \notin C \right\}$ is called the *set of the generating defaults* for the rational extension (E, C).

From the above definitions we can express the applicability condition of the generating defaults, in terms of derivability and non-derivability in classical logic as follows:

- $\frac{\alpha:\beta}{\gamma}$ is a generating default for a *classical extension:* E if its prerequisite is derivable from the actual extension: $E \vdash \alpha$ and the negation of its justification is not derivable from the *corresponding extension*: $E \nvdash \neg\beta$.
- $\frac{\alpha:\beta}{\gamma}$ is a generating default for a *rational extension:* (E, C) if its prerequisite is derivable from the actual extension: $E \vdash \alpha$ and the negation of its justification is not derivable from the *corresponding reasoning context*: $C \nvdash \neg\beta$.

The applicability condition for classical default logic is an individual one and for rational default logic is a global one, taking in consideration all the justifications (memorized in the context) used in the reasoning process.

Example 2.1. The default theory *(D,W)* with $W = \{F \vee C\}$ and $D = \left\{d1 = \frac{:A}{B}, d2 = \frac{:\neg A}{C}, d3 = \frac{:\neg B \wedge \neg F}{G}, d4 = \frac{:\neg B \wedge \neg C}{E}\right\}$ has:

- **one classical default extension**:

$E1 = Th(\{F \vee C, B, C\})$ with $D1 = \{d1, d2\}$ as generating default set.

- **two rational default extensions**:

1. $(E2, C2) = (Th(\{F \vee C, B\}), Th(\{F \vee C, B, A\}))$ generated by $D2 = \{d1\}$;
2. $(E3, C3) = (Th(\{F \vee C, C, G\}), Th(\{F \vee C, G, \neg A, \neg B \wedge \neg F\}))$ generated by the set $D3 = \{d2, d3\}$.

We remark that:

- $F \vee C$, B,C,$B \wedge C$,$B \vee C$, $F \vee C \vee B$, $F \wedge C$ are *credulous* and also *skeptical classical* default consequences;
- $F \vee C$, $B \vee C$, $F \vee C \vee G$, $F \vee C \vee B$, $B \vee G$ are *skeptical rational* default consequences belonging to both rational extensions;
- all skeptical rational consequences are also credulous rational consequences;

- B, C, G, $C \vee G$, $C \wedge G$, $F \wedge C$, $(F \vee C) \wedge G$ are *credulous* (but not skeptical) *rational* default consequences belonging to at least one of the rational extensions.

3. Sequent and anti-sequent calculi in propositional logic

The *sequent calculus*, as an improvement of Gentzen natural deduction system, is an axiomatization of classical logic and also provides a direct and syntactic proof method. The *anti-sequent calculus* for propositional logic was introduced in [3] as a complementary system of sequent calculus.

These two axiomatic systems are used to check the derivability and non-derivability in propositional logic.

A *sequent* has the form: $U \Rightarrow V$ and an *anti-sequent* has the form $U \not\Rightarrow V$, where U and V are finite sets of propositional formulas. U is called *antecedent* and V is called *succedent*.

A *basic sequent* contains the same formula, A, in both antecedent and succedent: $U, A \Rightarrow V, A$ or has the form $U \Rightarrow true$.

An anti-sequent $U \not\Rightarrow V$ is called a *basic anti-sequent* if all the formulas of U and V are atomic formulas and $U \cap V = \emptyset$.

Semantics

- The sequent $U \Rightarrow V$ is *true* if each model of U is also a model for at least one of the formulas of V.
- $U \not\Rightarrow V$ is *true* if there is a model M of U in which all the formulas of V are false. M is called an *anti-model* for this anti-sequent.

Axioms

- All basic sequents are true, therefore they are the *axioms* of sequent calculus.
- The basic anti-sequents are true and represent the *axioms* of anti-sequent system.

The rules of sequent calculus and anti-sequent calculus are presented in TABLE 1 and TABLE 2. These rules can be applied as *inference rules*: from the premises (sequents/ anti-sequent above the line) to the conclusion (sequent/anti-sequent below the line) or backwards from the conclusion to the premises as *reduction rules*.

The theorems of soundness and completness for these two systems can be expressed in an uniform manner as follows:

Theorem 3.1.
A sequent/an anti-sequent is true if and only if it can be reduced to basic sequents/anti-sequent using the reduction rules.

From TABLE 1 and TABLE 2 we remark that the rules with two premisses of sequent calculus are splitted into pairs of rules in anti-sequent calculus. Thus the exhaustive search in sequent calculus becomes nondeterminism in anti-sequent calculus and the reduction process is a linear one.

TABLE 1. Sequent and anti-sequent *left* rules - *Introduction into antecedent*

Connective	Sequent rules	Anti-sequent rules
$\neg$	$(\neg_l)\ \dfrac{U \Rightarrow V, A}{U, \neg A \Rightarrow V}$	$(\neg^c{}_l)\ \dfrac{U \not\Rightarrow V, A}{U, \neg A \not\Rightarrow V}$
$\wedge$	$(\wedge_l)\ \dfrac{U, A, B \Rightarrow V}{U, A \wedge B, \Rightarrow V}$	$(\wedge^c{}_l)\ \dfrac{U, A, B \not\Rightarrow V}{U, A \wedge B, \not\Rightarrow V}$
$\vee$	$(\vee_l)\ \dfrac{U, A \Rightarrow V \quad U, B \Rightarrow V}{U, A \vee B, \Rightarrow V}$	$(\vee^c{}_{l1})\ \dfrac{U, A \not\Rightarrow V}{U, A \vee B \not\Rightarrow V}$ $(\vee^c{}_{l2})\ \dfrac{U, B \not\Rightarrow V}{U, A \vee B \not\Rightarrow V}$
$\rightarrow$	$(\rightarrow_l)\ \dfrac{U \Rightarrow A, V \quad U, B \Rightarrow V}{U, A \rightarrow B \Rightarrow V}$	$(\rightarrow^c{}_{l1})\ \dfrac{U \not\Rightarrow A, V}{U, A \rightarrow B \not\Rightarrow V}$ $(\rightarrow^c{}_{l2})\ \dfrac{U, B \not\Rightarrow V}{U, A \rightarrow B \not\Rightarrow V}$

TABLE 2. Sequent and anti-sequent *right* rules - *Introduction into succedent*

Connective	Sequent rules	Anti-sequent rules
$\neg$	$(\neg_r)\ \dfrac{U, A \Rightarrow V}{U \Rightarrow V, \neg A}$	$(\neg^c{}_r)\ \dfrac{U, A \not\Rightarrow V}{U \not\Rightarrow V, \neg A}$
$\wedge$	$(\wedge_r)\ \dfrac{U \Rightarrow A, V \quad U \Rightarrow B, V}{U \Rightarrow A \wedge B, V}$	$(\wedge^c{}_{r1})\ \dfrac{U \not\Rightarrow A, V}{U \not\Rightarrow A \wedge B, V}$ $(\wedge^c{}_{r2})\dfrac{U \not\Rightarrow B, V}{U \not\Rightarrow A \wedge B, V}$
$\vee$	$(\vee_r)\ \dfrac{U \Rightarrow A, B, V}{U \Rightarrow A \vee B, V}$	$(\vee^c{}_r)\ \dfrac{U \not\Rightarrow A, B, V}{U \not\Rightarrow A \vee B, V}$
$\rightarrow$	$(\rightarrow_r)\ \dfrac{U, A \Rightarrow B, V}{U \Rightarrow A \rightarrow B, V}$	$(\rightarrow^c{}_r)\ \dfrac{U, A \not\Rightarrow B, V}{U \not\Rightarrow A \rightarrow B, V}$

The *derivability* in propositional logic is expressed in sequent calculus as follows:
$U1, U2, \ldots, Un \vdash V1 \vee V2 \vee \ldots \vee Vm$ *if and only if*
the sequent $U1, U2, \ldots, Un \Rightarrow V1, V2, \ldots, Vm$ is **true**,
meaning that from the conjunction of the hypothesis at least one of the formulas from the succedent can be proved.

The *non-derivability* in propositional logic is expressed in anti-sequent calculus as follows:
$U1, U2, \ldots, Un \nvdash V1 \wedge V2 \wedge \ldots \wedge Vm$ *if and only if*
the anti-sequent $U1, U2, \ldots, Un \not\Rightarrow V1, V2, \ldots, Vm$ is **true**,
meaning that from the conjunction of the hypothesis none of the formulas from the succedent can be proved.

The following theorem shows the complementarity of these two systems:

Theorem 3.2.([4])
The anti-sequent $U \not\Rightarrow V$ *is true* if and only if the sequent $U \Rightarrow V$ *is not true.*

These two axiomatic systems will be used in the following section to check the applicability conditions of the defaults: the derivability of the prerequisites and the non-derivability of the justifications.

4. Axiomatization of credulous reasoning in rational default logic

Based on the credulous sequent calculus for classical default logic proposed in [3], in this section we introduce an abstract characterization of the credulous nonmonotonic inference in rational default logic. An axiomatic system called *credulous rational default sequent calculus* is introduced.

Definition 4.1. Let (D, W) be a propositional default theory.
A *credulous rational default sequent* has the *syntax*:

$$(Pre, Just); (W, D, Just_c) \longmapsto U.$$

U is a set of propositional formulas and is called *succedent*. The *antecedent* contains two components:

- the first component represented by Pre and $Just$ contains constraints regarding the prerequisites and the justifications of the defaults. The constraints are expressed using the modalities M (possibility) and L (necessity).
- the second component is composed of W, D representing the propositional default theory and $Just_c$ containing the justifications assumed to be true during the reasoning process. In this variant of the default logic we need $Just_c$ in order to check the global applicability condition of the justifications of the defaults.

Remarks:
A constraint of the form $M\alpha$ is satisfied by a set E of sentences if $\neg\alpha$ is not derivable from E, and this can be expressed in anti-sequent calculus as: $E \not\Rightarrow \neg\alpha$.
A set E of sentences satisfies a constraint of the form $L\delta$ if δ is derivable from E, expressed as: $E \Rightarrow \delta$ in sequent calculus.

Definition 4.2. The *semantics* of a credulous rational default sequent:
The *credulous rational sequent* $(Pre, Just); (W, D, Just_c) \longmapsto U$ *is true* if $\vee U$ belongs to at least one rational extension of the theory (W, D), that satisfies the constraints from Pre and $Just$ and is guided by the reasoning context $Th(W \cup Just_c)$.

Definition 4.3. The *axiomatic system* is $Cr = (\Sigma_{Cr}, F_{Cr}, A_{Cr}, R_{Cr})$, where:

- Σ_{Cr} contains all the symbols used to build propositional formulas, modal propositional formulas and defaults defined on the underlying language of the default theory.

- F_{Cr} contains all classical sequents, all classical anti-sequents and all credulous rational default sequents defined on the underlying language of the default theory.
- A_{Cr}, the set of axioms of this formal system, contains all the basic sequents and basic anti-sequents defined on the underlying language of the default theory.
- R_{Cr} = reduction rules = $\{sequent\ rules, anti-sequent\ rules\} \cup$ $\cup \{R1, R2, R3, R4, R5, R6, R7, R8\}$

We propose the following specific reduction rules based on Definition 2.2.

Sequent rules for rational default logic:

(R1) $$\frac{W \Rightarrow U}{(\emptyset, \emptyset)\ ;\ (W, D, \emptyset)\ \longmapsto U}$$

(R2) $$\frac{W \Rightarrow U}{(\emptyset, \emptyset)\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}$$

(R3) $$\frac{W \Rightarrow \alpha \qquad (Pre, Just \cup \{M\beta\})\ ;\ (W \cup \{\gamma\}, D, Just_c \cup \{\beta\})\ \longmapsto U}{(Pre, Just); (W, D \cup \left\{\frac{\alpha:\beta}{\gamma}\right\}, Just_c)\ \longmapsto U}$$

(R4) $$\frac{(Pre \cup \{M\neg\alpha\}, Just)\ ;\ (W, D, Just_c)\ \longmapsto U}{(Pre, Just)\ ;\ (W, D \cup \left\{\frac{\alpha:\beta}{\gamma}\right\}, Just_c)\ \longmapsto U}$$

(R5) $$\frac{(Pre, Just \cup \{L\neg\beta\})\ ;\ (W, D, Just_c)\ \longmapsto U}{(Pre, Just)\ ;\ (W, D \cup \left\{\frac{\alpha:\beta}{\gamma}\right\}, Just_c)\ \longmapsto U}$$

(R6) $$\frac{W \not\Rightarrow \alpha \qquad (Pre, Just)\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}{(Pre \cup \{M\neg\alpha\}, Just)\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}$$

(R7) $$\frac{W \cup Just_c \not\Rightarrow \neg\beta \qquad (\emptyset, Just)\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}{(\emptyset, Just \cup \{M\beta\})\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}$$

(R8) $$\frac{W \cup Just_c \Rightarrow \neg\beta \qquad (\emptyset, Just)\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}{(\emptyset, Just \cup \{L\neg\beta\})\ ;\ (W, \emptyset, Just_c)\ \longmapsto U}$$

Remarks:

- The rule $R1$ shows that default logic extends the classical logic: if the succedent is derivable from the set of facts (W), it can be deduced from the whole default theory also.
- When all the defaults were introduced (as applicable or non-applicable) and the corresponding constraints were checked ($Pre = \emptyset$, $Just = \emptyset$) then the default rational sequent is reduced to a classical one using $R2$.
- In the reasoning process the introduction of an applicable default, $d = \frac{\alpha:\beta}{\gamma}$, is formalized by the rule $R3$.

-*First premise*: the derivability of the premise α is checked using the classical sequent calculus.

-*Second premise*: the justification β is added to $Just_c$, the consequent γ is added to the set of facts and the corresponding constraint $M\beta$ for justification is introduced.

- The rules $R4$ and $R5$ are used to introduce a default $\frac{\alpha:\beta}{\gamma}$ as non-applicable either by considering its prerequisite as non-derivable (constraint: $M\neg\alpha$) or its justification inconsistent in the context (constraint: $L\neg\beta$).
- $R6$ is used to check the constraints (M) corresponding to the prerequisites of the non-applicable defaults.
- Applying the rules $R7/R8$, the constraints (M/L) corresponding to the justifications are checked in the reasoning context, using anti-sequent/ sequent calculus.
- The order of applying the specific reduction rules is as follows:
  - the rules $R3$, $R4$ and $R5$ are used for introducing all the defaults as applicable or non-applicable until $D = \emptyset$;
  - $R6$ is applied to check the constraints corresponding to the prerequisites until $Pre = \emptyset$;
  - the constraints for justifications are checked using the rules $R7$ and $R8$ until $Just = \emptyset$;
  - when $D = \emptyset, Pre = \emptyset$ and $Just = \emptyset$, the default sequent is reduced to a classical one using $R2$;
  - the classical sequents/anti-sequents are further reduced using the reduction rules from sequent/anti-sequent calculus.
- From the point of view of the classical default logic the above axiomatic system is a reformulation of the one proposed in [3], if we eliminate $Just_c$.

Theorem 4.1. The credulous rational default sequent calculus is *sound and complete*: a credulous sequent is derivable if and only if it is true (can be reduced to classical basic sequents and basic anti-sequents).

Proof: The proof is based on the proof from [3] which can be easily adapted for rational default logic using Definition 2.2.

Consequence:

A formula X is a credulous rational default consequence of the default theory (D, W) if the credulous rational sequent $(\emptyset, \emptyset); (W, D, \emptyset) \longmapsto X$ is true.

Example 4.1. We will show that the formula $C \wedge G$ is a credulous rational belief of the default theory (D, W) from Example 2.1.

$W = \{F \vee C\}$ and $D = \left\{d1 = \frac{:A}{B}, d2 = \frac{:\neg A}{C}, d3 = \frac{:\neg B \wedge \neg F}{G}, d4 = \frac{:\neg B \wedge \neg C}{E}\right\}$.

The reasoning process modeled by the credulous rational default sequent calculus is represented by the following up-side-down binary tree.

$$\frac{S6{:}\overline{F \vee C, C, G \Rightarrow G} \qquad S7{:}\overline{F \vee C, C, G \Rightarrow C}}{- - - - - - - - - - - - - - - - - -}\ \wedge_r$$

$$F\vee C, C, G \Rightarrow G\wedge C$$
$$\text{- - - - - - - - - - - - - - - - - -}\ R2$$
$$S5 \quad (\emptyset,\emptyset);(\{F\vee C,C,G\},\emptyset,\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R7$$
$$S4 \quad (\emptyset,\{M(\neg B\wedge\neg F)\});(\{F\vee C,C,G\},\emptyset,\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R7$$
$$S3 \quad (\emptyset,\{M\neg A,M(\neg B\wedge\neg F)\});(\{F\vee C,C,G\},\emptyset,\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R8$$
$$S2 \quad (\emptyset,\{M\neg A,M(\neg B\wedge\neg F),L\neg(\neg B\wedge\neg C\});(\{F\vee C,C,G\},\emptyset,\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R8$$
$$(\emptyset,\{M\neg A,M(\neg B\wedge\neg F),L\neg A,L\neg(\neg B\wedge\neg C\});(\{F\vee C,C,G\},\emptyset,\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R5$$
$$(\emptyset,\{M\neg A,M(\neg B\wedge\neg F),L\neg A\});(\{F\vee C,C,G\},\left\{d4=\tfrac{:\neg B\wedge\neg C}{E}\right\},\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R5$$
$$S1 \quad (\emptyset,\{M\neg A,M(\neg B\wedge\neg F)\});(\{F\vee C,C,G\},\left\{\tfrac{:A}{B},\tfrac{:\neg B\wedge\neg C}{E}\right\},\{\neg A,\neg B\wedge\neg F\}) \longmapsto G\wedge C$$
$$\text{- -}\ R3$$
$$S1 \quad (\emptyset,\{M\neg A\});(\{F\vee C,C\},\left\{d1=\tfrac{:A}{B},d3=\tfrac{:\neg B\wedge\neg F}{G},d4=\tfrac{:\neg B\wedge\neg C}{E}\right\},\{\neg A\}) \longmapsto G\wedge C$$
$$\text{- -}\ R3$$
$$(\emptyset,\emptyset);(\{F\vee C\},\left\{d1=\tfrac{:A}{B},d2=\tfrac{:\neg A}{C},d3=\tfrac{:\neg B\wedge\neg F}{G},d4=\tfrac{:\neg B\wedge\neg C}{E}\right\},\emptyset) \longmapsto G\wedge C$$

- The initial credulous rational default sequent is:
  $(\emptyset,\emptyset);(\{F\vee C\},\left\{\frac{:A}{B},\frac{:\neg A}{C},\frac{:\neg B\wedge\neg F}{G},\frac{:\neg B\wedge\neg C}{E}\right\},\emptyset) \longmapsto G\wedge C$
- The defaults $d2$ and $d3$ are introduced as applicable defaults using $R3$. Their prerequisites are checked in the first premise $S1$: $F\vee C \Rightarrow true$ which is a classical basic sequent. The corresponding M-constraints for their justifications are introduced in the second premise.
- Using twice the rule $R5$, $d1$ and $d4$ are considered non-applicable defaults and thus L-constraints are added to the set of constraints.
- All the defaults were either introduced as applicable or non-applicable and we used the rules $R7$ and $R8$ to check the satisfiability of their corresponding constraints (L and M) for justifications in the reasoning context: $Th(\{F\vee C,G,C\}\cup\{\neg A,\neg B\wedge\neg F\})$.
- The premises $S2$ and $S3$ are classical basic sequents:
  $S2: F\vee C,G,C,\neg A,\neg B\wedge\neg F\Rightarrow\neg A$,
  $S3: F\vee C,G,C,\neg A,\neg B\wedge\neg F\Rightarrow B\vee C$
- The premises $S4$ and $S5$ are classical basic anti-sequents:
  $S4: F\vee C,G,C,\neg A,\neg B\wedge\neg F\not\Rightarrow A$,
  $S5: F\vee C,G,C,\neg A,\neg B\wedge\neg F\not\Rightarrow B\vee F$
- $R2$ reduces the credulous rational sequent (with $D=\emptyset$, $Pre=\emptyset$, $Just=\emptyset$), to a classical sequent which can be further reduced to basic sequents ($S6$ and $S7$) using $\wedge_r$.

The initial credulous rational default sequent was reduced to basic sequents($S1$, $S2$,$S3$,$S6$,$S7$) and basic anti-sequents ($S4$, $S5$), therefore it is a true sequent and according to the soundness of our axiomatic system we have that the formula $C\wedge G$ is a credulous rational default consequence of the default theory (D,W).

$C \wedge G$ belongs only to the actual rational extension $E3 = Th(\{F \vee C, C, G\})$ generated by $D3 = \{d2, d3\}$. The corresponding reasoning context is $C3 = Th(\{F \vee C, G, \neg A, \neg B \wedge \neg F\}$.

5. Conclusions and further work

In this paper we introduced an axiomatic system for credulous reasoning in rational default logic. The proposed system, using specific reduction rules, reduces the nonmonotonic inferential process to a classic inferential process modelled by the sequent calculus and anti-sequent calculus for propositional logic.

As further work we propose an uniform axiomatization of credulous/skeptical reasoning, using sequent calculus, for all the versions (classical, justified, constrained and rational) of default logic. Also of great practical interest is to add to these axiomatic systems proof strategies in order to obtain efficient proof methods.

References

[1] Antoniou, G., Courtney, A.P., Ernst, J., Williams, M.A., "A System for Computing Constrained Default logic Extensions", Logics in Artificial Intelligence, Lecture Notes in Artificial Intelligence, Vol.1126, 1996, pp. 237–250.

[2] Antoniou, G., "Nonmonotonic reasoning", MIT Press, 1998.

[3] Bonatti, P.A,"Sequent Calculi for default and autoepistemic logics"", Proceedings of TABLEAUX'96, LNAI 1071, pp. 127–142, Springer-Verlag, Berlin, 1996.

[4] Bonatti, P.A., Olivetti, N., "Sequent Calculi for Propositional Nonmonotonic Logics", ACM Trans. Comput. Log., 2002, pp. 226–278.

[5] Cholewinski, P., Marek, W., Truszczynski, M., "Default reasoning system DeReS", Proceedings of KR-96, Morgan Kaufmann, 1996, pp. 518–528.

[6] Lukasiewicz, W., "Considerations on default logic - an alternative approach", Computational Intelligence **4**, 1988, pp. 1–16.

[7] Lupea M., "Nonmonotonic reasoning using default logics", Ph.D.Thesis,"'Babes-Bolyai"' University, Cluj-Napoca, 2002.

[8] Lupea M., "Nonmonotonic inference operations for default logics", ECIT - Symposium on Knowledge-based Systems and Expert Systems, Iasi, Romania, 2002, pp. 1-12.

[9] Marek, W., Truszczynski, M., "Normal form results for default logics", Non-monotonic and Inductive logic, LNAI Vol. 659, Springer Verlag, 1993, pp. 153–174.

[10] Milnikel, R.S., "Sequent calculi for skeptical reasoning in predicate default logic and other nonmonotonic logics", pp. 1-40, Kluwer, 2004.

[11] Mikitiuk, A., Truszczynski, M., "Rational default logic and disjunctive logic programming", Logic programming and non-monotonic reasoning, MIT Press, 1993, pp. 283–299.

[12] Nicolas, P., Saubion, F., Stephan, I., "Genetic algorithm for extension search in default logic", The 8-th International Workshop on Non-Monotonic Reasoning, 2000.

[13] Reiter, R., "A Logic for Default reasoning", Artificial Intelligence **13**, 1980, pp. 81–132.

[14] Schaub, T.H., "Considerations on default logics", Ph.D. Thesis, Technischen Hochschule Darmstadt, Germany, 1992.

[15] Schaub, T.H., "The automation of reasoning with incomplete information", Springer-Verlag, Berlin, 1997.

[16] Stalnaker, R.C., "What is a non-monotonic consequence relation", Fundamenta Informaticae, 21, 1995,pp 7–21.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania

E-mail address: lupea@cs.ubbcluj.ro

STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

FINAL NOTE ON DĂNUŢ MARCU

THE EDITORS

In a past issue of our journal, Studia Universitatis Babeş-Bolyai, Series Informatica, no. 1/2004, we have published a note on the plagiate papers of Mr. Dănuţ Marcu.

At that moment we have stated that, consequently to these plagiates, we decided to retract the papers of Mr. Marcu. As well, we mentioned that we have lost the confidence in Mr. Marcu, and that we decided to ban Mr. Marcu from ever publishing in our journal.

We would like to confirm once again our decision, that refers to retracting all the papers Dănuţ Marcu published in our journal.

This note is the final note on Mr. Marcu. We are hereby closing his case.

The Editors

Faculty of Mathematics and Computer Science, Department of Computer Science, Babes-Bolyai University, 400084 Cluj-Napoca, Romania

E-mail address: studia-i@cs.ubbcluj.ro

Received by the editors: July 15, 2007.