INFORMATION SCIENCE & COMPUTING

International Book Series Number 1

Algorithmic and Mathematical Foundations of the Artificial Intelligence

Supplement to
International Journal "Information Technologies and Knowledge" Volume 2 / 2008

ITHEA
SOFIA, 2008

Krassimir Markov, Krassimira Ivanova, Ilia Mitov (ed.)

Algorithmic and Mathematical Foundations of the Artificial Intelligence

International Book Series "INFORMATION SCIENCE & COMPUTING", Number 1

Supplement to the International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Volume 2 / 2008

Institute of Information Theories and Applications FOI ITHEA

Sofia, Bulgaria, 2008

This issue contains a collection of papers in the field of algorithmic and mathematical foundations of the Artificial Intelligence. Papers are selected from the International Conferences of the Joint International Events of Informatics "ITA 2008", Varna, Bulgaria.

International Book Series "INFORMATION SCIENCE & COMPUTING", Number 1
Supplement to the International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Volume 2, 2008

Edited by **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria,
in collaboration with
- **V.M.Glushkov Institute of Cybernetics of NAS**, Ukraine,
- **Institute of Mathematics and Informatics, BAS**, Bulgaria,
- **Institute of Information Technologies, BAS**, Bulgaria.

Publisher: Institute of Information Theories and Applications FOI ITHEA, Sofia, 1000, P.O.B. 775, Bulgaria.
Издател: Институт по информационни теории и приложения ФОИ ИТЕА, София, 1000, п.к. 775, България
www.ithea.org, www.foibg.com, e-mail: info@foibg.com

General Sponsor: **Consortium FOI Bulgaria** (www.foibg.com).

Printed in Bulgaria

ISSN 1313-0455 (printed) **ISSN 1313-048X (online)** **ISSN 1313-0501 (CD/DVD)**

PREFACE

The scope of the International Book Series “Information Science and Computing” (**IBS ISC**) covers the area of Informatics and Computer Science. It is aimed to support growing collaboration between scientists from all over the world. IBS ISC is official publisher of the works of the members of the ITHEA International Scientific Society.

The official languages of the IBS ISC are English and Russian.

IBS ISC welcomes scientific papers and books connected with any information theory or its application.

IBS ISC rules for preparing the manuscripts are compulsory. The rules for the papers and books for IBS ISC are given on www.foibg.com/ibsisc. The camera-ready copy of the papers and books should be received by e-mail: info@foibg.com.

Responsibility for papers and books published in IBS ISC belongs to authors.

The Number 1 of the IBS ISC contains collection of papers from the field of Algorithmic and Mathematical Foundations of the Artificial Intelligence. Papers are peer reviewed and are selected from the several International Conferences, which were part of the Joint International Events of Informatics “ITA 2008”, Varna, Bulgaria.

ITA 2008 has been organized by

Institute of Information Theories and Applications FOI ITHEA

in collaboration with:

- ITHEA International Scientific Society
- International Journal “Information Theories and Applications”
- International Journal "Information Technologies and Knowledge"
- Association of Developers and Users of Intelligent Systems (Ukraine)
- Association for Development of the Information Society (Bulgaria)
- V.M.Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine
- Institute of Mathematics and Informatics, BAS (Bulgaria)
- Institute of Information Technologies, BAS (Bulgaria)
- Institute of Mathematics of SD RAN (Russia)
- Taras Shevchenko National University of Kiev (Ukraine)
- Universidad Politecnica de Madrid (Spain)
- BenGurion University (Israel)
- Rzeszow University of Technology (Poland)
- University of Calgary (Canada)
- University of Hasselt (Belgium)
- Kharkiv National University of Radio Electronics (Ukraine)
- Astrakhan State Technical University (Russia)
- Varna Free University “Chernorizets Hrabar” (Bulgaria)
- National Laboratory of Computer Virology, BAS (Bulgaria)
- Uzhgorod National University (Ukraine)
- Sofia University “Saint Kliment Ohridski” (Bulgaria)
- Technical University – Sofia (Bulgaria)
- New Bulgarian University (Bulgaria)

The main ITA 2008 events were:

KDS	XIVth International Conference "Knowledge - Dialogue – Solution"
i.Tech	Sixth International Conference "Information Research and Applications"
MeL	Third International Conference "Modern (e-) Learning"
ISK	Second International Scientific Conference "Informatics in the Scientific Knowledge"
INFOS	International Conference "Intelligent Information and Engineering Systems"
GIT	Sixth International Workshop on General Information Theory
CS	Third International Workshop "Cyber Security"
eM&BI	Second International Workshop "e-Management & Business Intelligence"
IMU ICT	International Seminar "Information Models' Utility in Information and Communication Technologies"
ISSI	Second International Summer School on Informatics

More information about ITA 2008 International Conferences is given at the www.foibg.com.

The great success of ITHEA International Journals, International Book Series and International Conferences belongs to the whole of the ITHEA International Scientific Society.

We express our thanks to all authors, editors and collaborators who had developed and supported the International Book Series "Information Science and Computing".

General Sponsor of IBS ISC is the **Consortium FOI Bulgaria** (www.foibg.com).

Sofia, June 2008 *Kr. Markov, Kr. Ivanova, I. Mitov*

TABLE OF CONTENTS

Papers in English

Papers in Russian

INDEX OF AUTHORS

NON-LINEAR NETWORK-FLOW MODEL OF ŁUKASIEWICZ'S MULTIVALUE LOGIC

Vassil Sgurev, Stefan Kojnov

***Abstract**: The paper presents a new network-flow interpretation of Łukasiewicz's logic based on models with an increased effectiveness. The obtained results show that the presented network-flow models principally may work for multivalue logics with more than three states of the variables i.e. with a finite set of states in the interval from 0 to 1. The described models give the opportunity to formulate various logical functions. If the results from a given model that are contained in the obtained values of the arc flow functions are used as input data for other models then it is possible in Łukasiewicz's logic to interpret successfully other sophisticated logical structures. The obtained models allow a research of Łukasiewicz's logic with specific effective methods of the network-flow programming. It is possible successfully to use the specific peculiarities and the results pertaining to the function 'traffic capacity of the network arcs'. Based on the introduced network-flow approach it is possible to interpret other multivalue logics – of E.Post, of L.Brauer, of Kolmogorov, etc.*

***Keywords**: Łukasiewicz's multivalue logic, operational research, network flow interpretation.*

***ACM Classification Keywords**: F.4.0 Mathematical Logic and Formal Languages – General, F.4.1 Mathematical Logic – Logic and constraint programming.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

Most often the popular publication of J.Łukasiewicz [1] is accepted as a beginning of the multivalue logic.

Together with the classic logical systems since the middle of the past century different models of multivalue logic were an object of significant development [2].

During the second half of the same century a research activity started to use precise quantitative methods from the operational research to describe various logical operations. To achieve this goal most widely were used the methods of mixed integer programming (MIP) [4, 5].

A network-flow interpretation of operations and formulas from the propositional logic was introduced in [7] for decision-making systems. Due to its specific character in series of cases the network-flow methods lead to a greater effectiveness compared to MIP.

An attempt was made for a network-flow interpretation of Łukasiewicz's multivalue logic in [8]. The applied models were nonlinear with a significant degree of sophistication.

The paper presents a new network-flow interpretation of Łukasiewicz's logic based on models with an increased effectiveness.

Non-Linear Network-Flow Model of Łukasiewicz's Multivalue Logic

A new nonlinear network flow will be used that most generally can be defined in the following way [9]. For every $x_k \in X$

$$\sum_{i \in \Gamma_k^+} f_i - \sum_{j \in \Gamma_k^-} f_j = \begin{cases} v_k \ iff \ x_k \in S; \\ 0 \ \ iff \ x_k \notin \{S \cup T\}; \\ -v_k \ iff \ x_k \in T; \end{cases} \quad (1)$$

$$F_r(f_i, f_j) = f_r, \text{ where } i, j, r \in T; \quad (2)$$

$$0 \le f_i \le 1 \text{ for each arc } u_i \in U; \tag{3}$$

where $G(X,U)$ is a graph with a set of nodes *X* and a set of arcs *U*; f_i is a network function over the arc $u_i \in U$; *S* and *T* are respective sets of sources and consumers; c_i is the traffic capacity of the arc u_i; Γ_k^+ and Γ_k^- are the sets of indexes for all arcs that are the respective input and output for the node $x_k \in X$; v_k is the flow for the node $x_k \in S \bigcup T$; *T* is the set of indexes for the equalities (2).

It is assumed that the traffic capacities $\{c_i\}$ are integers and as a rule always equal to *1* while the arc flow functions $\{f_i\}$ may have various nonnegative values in the interval from *0* up to *1*.

In the propositional logic of Łukasiewicz the propositions may be in one of three possible states: true, false and neutral, respectively *1*, *0* and $1/2$.

The truth tables for disjunction and conjunction in Łukasiewicz's logic numerally have the following appearance:

Table 1 (f_3)

f_1 \ f_2	**1**	**0**	$1/2$
1	*1*	*1*	*1*
0	*1*	*0*	$1/2$
$1/2$	*1*	$1/2$	$1/2$

Table 2 (f_4)

f_1 \ f_2	**1**	**0**	$1/2$
1	*1*	*1*	$1/2$
0	*0*	*0*	*0*
$1/2$	*1*	$1/2$	$1/2$

If *A* and *B* are propositions and their respective numerical functions are f_1 and f_2 i.e.

$$A = f_1 \text{ and } B = f_2; \tag{4}$$

then from the two tables above it follows that for the disjunction $A \vee B$ and the conjunction $A \wedge B$ we may denote respectively

$$f_3 = max(f_1, f_2) \text{ and } f_4 = min(f_1, f_2). \tag{5}$$

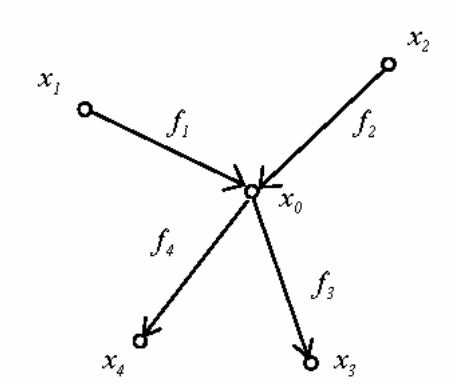


Fig. 1

The two logical operations will be interpreted by the following below subgraph and the corresponding to it flow equalities and inequalities:

$$f_1 + f_2 - f_3 - f_4 = 0; \tag{6}$$

$$k(f_1 - f_2) + f_2 = f_3; \tag{7}$$

$$f_1 \le f_3; \;\; f_2 \le f_3; \tag{8}$$

$$0 \le f_i \le 1; \;\; i = 1; 2; \ldots; 4 \text{ and } k = 0 \text{ or } 1 \tag{9}$$

where (6) is an equality for preservation in node x_0; (7) is a nonlinear dependency corresponding to (2) via which the disjunction from (5) is realized; (8) are additional linear constraints assisting the choice of the greater variable from f_1 or f_2.

The equations (1) for the nodes from x_1 to x_4 do not need to be explained because they are trivial: $v_i = f_i$ for $i \in \{1, 2\}$ and $v_j = f_j$ for $j \in \{3, 4\}$.

The variable f_3 takes the bigger value from f_1 and f_2: if $k = 1$ then the value is from f_1 and if $k = 0$ then the value is from f_2.

The dependencies from (5) to (9) serve the determination of the variables f_1 and f_2 with the bigger value i.e. these dependencies guarantee the requirements from Table 1.

The conjunction, the second equality from (5) may be realized via the same dependencies from (6) to (9) but the inequalities (8) must be replaced by the new dependencies

$$f_1 \geq f_4 \text{ and } f_2 \geq f_4 \tag{10}$$

assisting the choice of the smaller variable from f_1 or f_2.

On the other hand if the disjunction f_3 from (5) is already determined then it is possible to determine f_4 directly from the equation of preservation (6):

$$f_4 = f_1 + f_2 - f_3; \tag{11}$$

$$f_3 \leq f_1 + f_2 \text{ and } f_4 \leq f_1 + f_2 \tag{12}$$

Therefore the subgraph from Fig. 1 and the dependencies from (6) to (9) are ample factors to determine synonymously the disjunction $A \vee B = f_3$ and the conjunction $A \wedge B = f_4$ in Łukasiewicz's logic.

Negation in the same logic is determined by the functions:

$$\neg A = 1 - f_1 \text{ and } \neg B = 1 - f_2 \tag{13}$$

Implication in the logic of Łukasiewicz satisfies the requirement:

$$f_3 = \begin{cases} 1 \ \textit{iff} \ f_1 \leq f_2; \\ 1 - f_1 + f_2 \ \textit{iff} \ f_1 > f_2; \end{cases} \tag{14}$$

or otherwise

$$f_3 = min\left[1, \left(1 - f_1 + f_2\right)\right] \tag{15}$$

where

$$A = f_1; \ B = f_2 \text{ and } \left(A \rightarrow B\right) = f_3. \tag{16}$$

The truth table for implication by Łukasiewicz is of the following type:

Table 3 $\left(f_3\right)$

f_1 \ f_2	**1**	**0**	½
1	*1*	*1*	½
0	*1*	*1*	*1*
½	*1*	½	*1*

The data from this table may be juxtaposed to the subgraph from Fig. 1 and to the following dependencies for a network flow:

$$k\left(f_1 - f_2\right) + f_2 = f_3; \tag{17}$$

$$f_3 \leq 1 - f_1 + f_2 \text{ and } f_3 \leq 1; \tag{18}$$

with valid constraints (6) and (9).

In this flow the dependencies (17) and (18) guarantee the exact adherence to the equation (14). Here if $f_1 \leq f_2$ then the coefficient *k* has a value of zero and $f_3 = 1$ but in the opposite case *k* is unity and $f_3 = 1 - f_1 + f_2$.

In binary logic there exists an important equipollence:

$$A \rightarrow B \equiv \neg A \vee B \tag{19}$$

which as it is evident from Table 3 is not valid for all meanings of *A* and *B*. For example for $A = B = 1/2$ i.e. $f_1 = f_2 = 1/2$ then

$$A \to B = (1/2 \to 1/2) = 1 \qquad \text{(from Table 3);}$$

$$\neg A \vee B = (1 - 1/2) \vee 1/2 = 1/2 \quad \text{(from Table 2).}$$

An analogous conclusion can be made based on the network-flow realization of the two formulas from the equipollence (19). The first realization is shown via the dependencies (6), (9), (17) and (18) and the second formula from (19) may be interpreted by the disjunction from (6) to (9) where f_1 is replaced by its negation $1 - f_2$. Then

$$1 - f_1 + f_2 - f_3 - f_4 = 0;$$

$$k(1 - f_1 - f_2) + f_2 = f_3;$$

$$1 - f_1 \le f_3; \quad f_2 \le f_3.$$

The comparison of the last equalities and inequalities with the ones from (6), (17) and (18) shows that we have two different types of implications in the examined three-value logic, i.e. that the equipollence (19) is not valid (or it is not true) for all possible estimates in Łukasiewicz's logic.

The obtained results from (6) to (18) also show that the presented network-flow models principally may work for multivalue logics with more than three states of the variables i.e. with a finite set of states in the interval from *0* to *1*.

The described models give the opportunity to formulate various logical functions. If the results from a given model that are contained in the obtained values of the arc flow functions are used as input data for other models then it is possible in Łukasiewicz's logic to interpret successfully other sophisticated logical structures. The obtained models allow a research of Łukasiewicz's logic with specific effective methods of the network-flow programming. It is possible successfully to use the specific peculiarities and the results pertaining to the function 'traffic capacity of the network arcs'.

If we denote the value of some complex formula with f_3 i.e. $F(f_1, f_2, \ldots, f_k) = f_3$ while observing the equalities and the inequalities of the network flow respectively from (6) to (18) and formulating the goal functions $f_3 \to max$ and $f_3 \to min$ then it is possible to determine whether this formula is tautology and also what maximal and minimal values it may accept.

From the computational point of view the nonlinearity of the used models doubtlessly is a source of difficulties. The search of effective linear – precise and approximate network-flow methods and algorithms remains an important problem.

Based on the introduced network-flow approach it is possible to interpret other multivalue logics [2] – of E.Post, of L.Brauer, of Kolmogorov, etc.

Conclusion

The paper presents a new network-flow interpretation of Łukasiewicz's logic based on models with an increased effectiveness.

The obtained results show that the presented network-flow models principally may work for multivalue logics with more than three states of the variables i.e. with a finite set of states in the interval from 0 to 1.

The described models give the opportunity to formulate various logical functions.

If the results from a given model that are contained in the obtained values of the arc flow functions are used as input data for other models then it is possible in Łukasiewicz's logic to interpret successfully other sophisticated logical structures.

The obtained models allow a research of Łukasiewicz's logic with specific effective methods of the network-flow programming. It is possible successfully to use the specific peculiarities and the results pertaining to the function 'traffic capacity of the network arcs'.

Based on the introduced network-flow approach it is possible to interpret other multivalue logics – of E.Post, of L.Brauer, of Kolmogorov, etc.

Bibliography

[1] Cleene, S.C., Mathematical Logic, John Wiley and sons Inc., N.Y., 1967.

[2] Rescher, N., Many-valued Logic, McGrow Hill, N.Y., 1969.

[3] Łucasiewicz, J., Logica tzoiwartosciowa Ruch Filozofiezny, T.V, N.9, Lwow, 1920.

[4] Blair, C.E., R.G.Jeroslow, Some Results and Experiments in Programming Techniques for Propositional Logic, Comp. and Oper.Res., V.13, N.5, 1986, pp. 633-649.

[5] Hooker, J.N., A Quantitative Approach to Logic Inference, Decision Support Systems, N.4, 1988, pp. 45-69.

[6] Ford, L.R., D.R.Falkerson, Flow in Networks, Princeton University Press, 1962.

[7] Sgurev, V.S., Network Flow Approach Logic for Problem Solving, Int.J. on Information Theory and Appl., V.2, N.7, 1995, pp. 3-8.

[8] Nikolova, M., Network Flow Interpretation of Lukasiewicz Logic System, J. of Aut. And Informatique, UAI Publ., N.1, Sofia, 1998.

[9] Sgurev, V.S., Network Flow with General Constraints, BAS Publishing, Sofia, 1991.

Authors' Information

Vassil Sgurev *– Academician; Institute of Information Technologies, BAS, Acad. G.Bontchev St., bl.2, P.O.Box: 161, Sofia-1113, Bulgaria; e-mail: sgurev@bas.bg*

Stefan Kojnov *– Researcher, Institute of Information Technologies, BAS, Acad. G.Bontchev St., bl.29A, P.O. Box: 161, Sofia-1113, Bulgaria; e-mail: slk@iinf.bas.bg*

LAGRANGEAN APPROXIMATION FOR COMBINATORIAL INVERSE PROBLEMS

Hasmik Sahakyan, Levon Aslanyan

***Abstract**: Various combinatorial problems are effectively modelled in terms of (0,1) matrices. Origins are coming from n-cube geometry, hypergraph theory, inverse tomography problems, or directly from different models of application problems. Basically these problems are NP-complete. The paper considers a set of such problems and introduces approximation algorithms for their solutions applying Lagragean relaxation and related set of techniques.*

***Keywords**: Approximation algorithms, Lagragean relaxation*

***ACM Classification Keywords:** G.2.1 Discrete mathematics: Combinatorics*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

1. Introduction

A set of diverse combinatorial problems are defined and investigated in terms of discrete structures; in the simplest case these are $(0,1)$-matrices. Considered optimization problems come from specific subject areas – astronomy, medical diagnostics, seismology, etc. and are effectively modeled in terms of n-cube geometry, hypergraph degree sequences, image restoration, and other mathematical means. The common to these problems is that they look for inverses of some direct simple tasks. Most of these and related problems are NP-hard therefore approximate and heuristic algorithms are of interest. The area is studied intensively and we will brief in references [2-4].

Approximation algorithms introduced in this paper are based on Lagrangean relaxation and on variable splitting technique. These are well known widely implemented techniques of getting approximations. But in each particular case it is yet a question if the Lagrangean approach is effective. The problem under consideration is to be transformed into a form of Integer Linear Optimization with several groups of constraints. For each group of constraints it is necessary to have developed algorithms of their solutions. Finally it is yet a question whether the integration into Lagrangean framework will approach the optimal solution. To learn these possibilities a software environment is created for experimentations, which in addition provides solutions of Problems 1-3 considered below. As a demonstration example the Problem 2 is considered taking into account that differences between these problems are not critical.

Section 2 contains the necessary initial information and problem definitions. $(0,1)$-matrix interpretations are given in Section 3. In Section 4 Lagrangean relaxation method is applied to solve these problems. The splitting problems are given in Section 5 where Algorithms to solve the fragmental problems are constructed.

2. Problem descripion

We start with listing of minimal set of source problems.

P1. *n*-dimensional unit cube subsets with given partition (projection) sizes [4].

Vertices of n-dimensional unit cube is given by $E^n = \{(x_1, \cdots, x_n) / x_i \in \{0,1\}, i = \overline{1,n}\}$. Consider partition of E^n into the two subcubes $E^{n-1}_{x_i=1}$ and $E^{n-1}_{x_i=0}$ in accord to values of an arbitrary variable x_i. Similarly, each vertex subset $M \subset E^n$ can be partitioned into the $M_{x_i=1}$ and $M_{x_i=0}$.

Let M be an m-vertex subset of E^n. The vector $S=(s_1,\cdots,s_n)$ is called **associated (characteristic) vector of partitions** for the set M if $s_i=\left|M_{x_i=1}\right|$ for $i, 1\le i\le n$. The existence problem in this regard is to find out the existence of an m-vertex subset of E^n with given associated vector of partitions and a Boolean function by the given associated vector of activities. I

P2. Uniform hypergraphs with given Subsumed graphs' Degree Sequences

$k\ge 2$ is an integer and $V(G)$ is the vertex set with $\left|V(G)\right|\ge k$. Edges $E(G)$ of G are defined as members of $\begin{pmatrix} V(G) \\ k \end{pmatrix}$ which is the set of all k-subsets of $V(G)$. If G is k-hypergraph, $k\ge 3$ and $u\in V(G)$, then a $(k-1)$-hypergraph G_u is defines as follows. The vertex set $V(G_u)=V(G)-u$, and for each edge $B\in E(G)$ with $u\in V(G)$, $B-u$ is included as an edge of $E(G_u)$. We say that G *subsumes* the collection of hypergraphs $\{G_u / u\in V(G)\}$.

If G is a hypergraph and $u\in V(G)$, then $\deg(u)$ (degree of u), the number of edges containing u, is the number of edges of the subsumed graph G_u. The *degree sequence* of G is the multiset $DegSeq(G)=\{\deg(u)/u\in V(G)\}$.

Problem (Subsumed graphs' Degree Sequences). [3]

Given $DegSeq(g_i), i=1,\cdots,n$ of n graphs $g_1,\cdots,g_n$, is there an n vertex hypergraph G such that the subsumed graphs $G_1,\cdots,G_n$ satisfy $DegSeq(G_i)=DegSeq(g_i)$ for $i=1,\cdots,n$?

P3. Reconstruction of weighted $(0,1)$ -matrices [4].

The general image reconstruction problem is defined as follows: an image of $(m\times n)$ pixels of p different colors, has to be reconstructed. We are given the number $r(i,c)$ of pixels of each color c in each row i and also the number $s(j,c)$ of pixels of each color c in each column j; is it possible to reconstruct an image, for all i,j,c?

3. (0,1) matrix model of problems P1. P2. P3

Consider a $(0,1)$-matrix A of size $m\times n$. Let $R=(r_1,\cdots,r_m)$ and $S=(s_1,\cdots,s_n)$ denote the row and column sums of A respectively, and let $U(R,S)$ be the set of all $(0,1)$-matrices with row sums R and column sums S. A necessary and sufficient condition for the existence of a $(0,1)$ matrix of the class $U(R,S)$ was found by Gale and Ryser [R,1966].

We reformulate the basic problems P1, P2 and P3 in terms of $(0,1)$-matrices. Common to all problems are the given integer vectors $R=(r_1,\cdots,r_m)$ and $S=(s_1,\cdots,s_n)$.

Problem 1. Existence of a $(0,1)$ matrix with different rows in the class $U(R,S)$

Given vectors $R=(r_1,\cdots,r_m)$ and $S=(s_1,\cdots,s_n)$. Does there exist a matrix $X=\{x_{i,j}\}$ in the class $U(R,S)$ with different rows?

Problem 2. Existence of a $(0,1)$ matrix in the class $U(R,S)$, with given intersections of pairs of rows

Given $R=(r_1,\cdots,r_m)$, $S=(s_1,\cdots,s_n)$ and $R'=(r'_1,\cdots,r'_{C_m^2})$ vectors. Enumerate pairs of rows and let $p(i',i'')$ indicates the number of the pair (i',i'') for $1\le i'<i''\le m$. Then the problem is in existence of

a matrix $X=\{x_{i,j}\}$ in the class $U(R,S)$ with the following property: rows i' and i'' intersect (by 1's) in $r'_{p(i',i'')}$ places.

Problem 3. Existence of a $(0,1)$ matrix in the class $U(R,S)$ with given intersections of adjacent pairs of rows

Given vectors $R=(r_1,\cdots,r_m)$, $S=(s_1,\cdots,s_n)$ and $R'=(r'_1,\cdots,r'_{m-1})$. Does there exist a matrix $X=\{x_{i,j}\}$ in $U(R,S)$ with the given intersections of adjacent pairs of rows - rows i and $i+1$ intersect (by 1's) in exactly r'_i places ($i=1,\cdots,m-1$)?

Note . Consider rows i' and i'' and let $r_{i'}\le r_{i''}$. If rows are different, then their intersection size is less than $r_{i''}$. Assuming that $r_1\le\cdots\le r_m$, the requirement of different rows in Problem 1 can be replaced by the property: intersection size for all pairs of rows, (i',i''), $1\le i'<i''\le m$ is less than $r_{i''}$.

While the Problem 2 is NP-complete, the complexity of Problems 1 and 3 are not known: Problem 1 is a well known open problem [4]. Complexity issue of the Problem 3 is not addressed yet.

4. Integer linear programming formulations and Lagrangean relaxation formulas

Let X be a $(0,1)$-matrix of size $m\times n$. Enumerate pairs of rows and let $p(i',i'')$ indicates the number of the pair (i',i''), for $1\le i'<i''\le m$. For each pair of rows, (i',i''), we define n binary variables $y_{p(i',i''),j}$, such that. $(y_{p(i',i''),j}=1)\Leftrightarrow(x_{i',j}=1)\,\&\,(x_{i'',j}=1)$.

Obviously it can be provided by the following set of algebraic conditions:
$$\begin{cases} y_{p(i',i''),j}\le x_{i',j}\\ y_{p(i',i''),j}\le x_{i'',j}\\ y_{p(i',i''),j}\ge x_{i',j}+x_{i'',j}-1\end{cases}$$

Now Problems 1-3 above can be formulated in terms of integer linear programming. We focus only on Problem 2 giving the details for that case. Problems 1 and 3 can be reformulated as integer programming, then relaxed and solved, - by a similar way.

Recall that we assume $r_1\le\cdots\le r_m$.

Problem IP2 Given integer vectors $R=(r_1,\cdots,r_m)$, $S=(s_1,\cdots,s_n)$ and $R'=(r'_1,\cdots,r'_{C_m^2})$. The problem is in existence of an $m\times n$ binary matrix $X=\{x_{i,j}\}$ and a $(C_m^2)\times n$ binary matrix $Y=\{y_{i,j}\}$ such that

$$\text{(IP2)}\begin{cases}(1)\sum_{i=1}^{m}x_{i,j}=s_j,\ j=1,\cdots,n\\ (2)\sum_{j=1}^{n}x_{i,j}=r_i,\ i=1,\cdots,m\\ (3)\begin{cases}y_{p(i',i''),j}\le x_{i',j}\\ y_{p(i',i''),j}\le x_{i'',j}\\ y_{p(i',i''),j}\ge x_{i',j}+x_{i'',j}-1\end{cases}\quad 1\le i'<i''\le m,\ j=1,\cdots,n\\ (4)\sum_{j=1}^{n}y_{p(i',i''),j}=r'_{p(i',i'')}\quad 1\le i'<i''\le m\\ (5)x_{i,j}\in\{0,1\},\ y_{i,j}\in\{0,1\}\end{cases}$$

3. Lagrangean relaxation and variable splitting

In a way to solve this problem we apply the Lagrangean relaxation and variable splitting technique.

Lagrangean relaxation for integer linear programming

Consider the following optimization problem

(P) $Max_x\{fx / Ax \le b, Cx \le d, x \in Z\}$ in which some constraints are complicating (suppose $Ax \le b$), in the sense that one would be able to solve the same integer programming problem has these constraints not been present: $Max_x\{fx / Cx \le d, x \in Z\}$. One can take advantage of this situation by constructing a so-called Lagrangean relaxation in the following way. Let $\lambda \ge 0$ be a vector of multipliers and let (LR_λ) be the problem

(LR_λ) $Max_x\{fx + \lambda(b - Ax) / Cx \le d, x \in Z$. ($LR_\lambda$) is the Lagrangean relaxation of (P). Let $v(LR_\lambda)$ is the value of optimal solution of (LR_λ). The problem (LD) $Min_{\lambda \ge 0} v(LR_\lambda)$ is called the Lagrangean dual of (P) relative to the $Ax \le b$. The optimal value of LD is a smallest upper bound on the optimal value of (P). For Problem 2 we will use variable splitting technique - we split our problem into separate vertical and horizontal subproblems, then the horizontal subproblem is further separated into subproblems for each pair of rows. Thus we consider Lagrangean relaxation of (IP2). We duplicate variables $x_{i,j}$, getting 2 independent sets of variables $x^h_{i,j}$ and $x^v_{i,j}$, and then dualize the copy (duplication) constraint using Lagrangean multipliers $\lambda_{i,j}$.

$$
\text{(IP2LR)}\begin{cases}
\max\{\sum \lambda_{i,j}(x^h_{i,j} - x^v_{i,j})\} \\
(1)\sum_{i=1}^{m} x^v_{i,j} = s_j, j = 1,\cdots,n \\
(2)\sum_{j=1}^{n} x^h_{i,j} = r_i, i = 1,\cdots,m \\
(3)\begin{cases} y_{p(i',i''),j} \le x^h_{i',j} \\ y_{p(i',i''),j} \le x^h_{i'',j} \\ y_{p(i',i''),j} \ge x^h_{i',j} + x^h_{i'',j} - 1 \end{cases} \quad 1 \le i' < i'' \le m,\ j = 1,\cdots,n \\
(4)\sum_{j=1}^{n} y_{p(i',i''),j} = r'_{p(i',i'')} \quad 1 \le i' < i'' \le m \\
(5) x^h_{i,j}, x^v_{i,j} \in \{0,1\},\ y_{i,j} \in \{0,1\}
\end{cases}
$$

Split the problem into sub problems – horizontal and vertical

$$
\text{(IP2-v)}\begin{cases}
\max\{\sum \beta_{i,j} x^v_{i,j}\} \\
(1)\sum_{i=1}^{m} x^v_{i,j} = s_j, j = 1,\cdots,n \\
(2) x^v_{i,j} \in \{0,1\}
\end{cases}
$$

$$\text{(IP2-h)}\begin{cases}\max\{\sum \alpha_{i,j}x^h_{i,j}\}\\ (1)\sum_{j=1}^{n} x^h_{i,j}=r_i,\quad i=1,\cdots,m\\ (2)\begin{cases} y_{p(i',i''),j}\le x^h_{i',j}\\ y_{p(i',i''),j}\le x^h_{i'',j} \qquad 1\le i'<i''\le m,\ j=1,\cdots,n\\ y_{p(i',i''),j}\ge x^h_{i',j}+x^h_{i'',j}-1\end{cases}\\ (3)\sum_{j=1}^{n} y_{p(i',i''),j}=r'_{p(i',i'')} \qquad 1\le i'<i''\le m\\ (4)x^h_{i,j}\in\{0,1\},\ y_{i,j}\in\{0,1\}\end{cases}$$

Using similar reasons IP2-h is split into subproblems for each pair of rows:

$$\text{(IP2-h1)}\begin{cases}\max\{\sum_{j=1}^{n}(\alpha'_j x'_j+\alpha''_j x''_j)\}\\ (1)\sum_{j=1}^{n} x'_j=r',\quad \sum_{j=1}^{n} x''_j=r''\\ (2)\begin{cases} y_j\le x'_j\\ y_j\le x''_j \qquad j=1,\cdots,n\\ y_j\ge x'_j+x''_j-1\end{cases}\\ (3)\sum_{j=1}^{n} y_j=r^*\\ (4)x'_j,x''_j\in\{0,1\},\ y_j\in\{0,1\},\ j=1,\cdots,n\end{cases}$$

Further we apply an iterative procedure to find the optimisation coefficients $\lambda_{i,j}$. On each iteration we consider C^2_m+1 separate subproblems (C^2_m horizontal and 1 vertical). Each horizontal subproblem is formulated as a parameterised set system problem.

4. Algorithms for solving subproblems for pairs of rows

(IP2-h1) is equivalent to the following problem.

Problem of Weighted Threads. Given 2 sets of weighted elements $X'=\{x'_1,\cdots,x'_n\}$ and $X''=\{x''_1,\cdots,x''_n\}$. $\alpha'_i\ge 0$ is the weight of $x'_i\in X'$, and $\alpha''_i\ge 0$ is the weight of $x''_i\in X''$. Given also positive integers $r', r'', r^*, r'\le r'', r^*<r''$. The problem is in finding subsets $\tilde{X}'\subseteq X'$ and $\tilde{X}''\subseteq X''$, such that: $\left|\tilde{X}'\right|=r'$ and $\left|\tilde{X}''\right|=r''$, and

1. $\sum_{x_i^{'} \in \tilde{X}^{'}} \alpha_i^{'} + \sum_{x_i^{''} \in \tilde{X}^{''}} \alpha_i^{''} \to \max$

2. $\left| \left\{ (x_i^{'}, x_i^{''}) / x_i^{'} \in \tilde{X}^{'}, x_i^{''} \in \tilde{X}^{''} \right\} \right| = r^*$

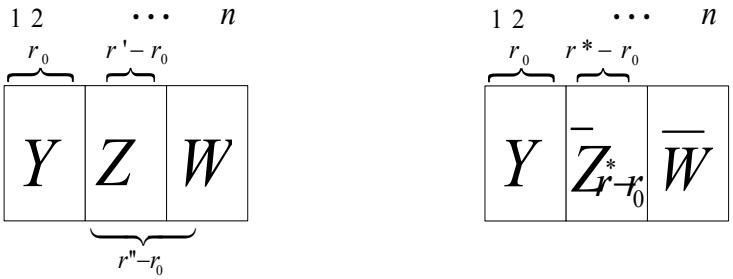
In its short description a three stage selection algorithm is constructed.

1. Arranging elements in $X^{'}$ and $X^{''}$ by decreasing order of their weights $\overrightarrow{X^{'}} = \{x_{i_1}^{'}, \cdots, x_{i_n}^{'}\}$, $x_{i_1}^{'} \geq \cdots \geq x_{i_n}^{'}$, and $\overrightarrow{X^{''}} = \{x_{j_1}^{''}, \cdots, x_{j_n}^{''}\}$, $x_{j_1}^{''} \geq \cdots \geq x_{j_n}^{''}$ and taking the first iteration for $\tilde{X}^{'}$ and $\tilde{X}^{''}$ as $X_{r^{'}}^{'} = \{x_{i_1}^{'}, \cdots, x_{i_{r^{'}}}^{'}\}$ and $X_{r^{''}}^{''} = \{x_{j_1}^{''}, \cdots, x_{j_{r^{''}}}^{''}\}$.

2. If $|Y| = r^*$ then $X_{r^{'}}^{'}$ and $X_{r^{''}}^{''}$ are the required subsets. Otherwise consider cases:

a) $|Y| = r_0 < r^*$ and b) $|Y| = r_0 > r^*$. It is enough to consider the first case:

Shift the elements of Y to the left.



Arrange elements (pairs) in $Z \cup W$ by decreasing order of sum of elements (weights) of the pair and denote by $\overline{Z}_{r^*-r_0}$ first $r^* - r_0$ elements, and by $\overline{W}$ - the reminder:

Construct the sets $\tilde{X}^{'}$ and $\tilde{X}^{''}$ by the elements of $Y \cup \overline{Z}_{r^*-r_0}$ (first element of each pair goes to $\tilde{X}^{'}$, second goes to $\tilde{X}^{''}$). Remaining $r^{'}-r^*$ elements of $\tilde{X}^{'}$ and $r^{''}-r^*$ elements of $\tilde{X}^{''}$ are formed as follows: Arrange elements by decreasing order in $\overline{W}^{'}$ and $\overline{W}^{''}$, where $\overline{W}^{'}$ and $\overline{W}^{''}$ consist of respectively the first (belonging to $X^{'}$) and second (belonging to $X^{''}$) elements of pairs of $\overline{W}$. Consider first $r^{'}-r^*$ subset in each set and denote them by $\overline{W}^{'}_{r^{'}-r^*}$ and $\overline{W}^{''}_{r^{'}-r^*}$. Subsets of remaining elements we denote by $\overline{W}^{'}_{rem}$ and $\overline{W}^{''}_{rem}$.

1. If there are no elements with the same index in $\overline{W}^{'}_{r^{'}-r^*}$ and $\overline{W}^{''}_{r^{'}-r^*}$, then these elements go to the $\tilde{X}^{'}$ and $\tilde{X}^{''}$ respectively.

2. Otherwise we replace the last element in subset $\overline{\overline{W}}'_{r'-r^*}$ by the first element of $\overline{\overline{W}}'_{rem}$ or replace the last element in $\overline{\overline{W}}''_{r'-r^*}$ by the first element of $\overline{\overline{W}}''_{rem}$ depending on the sum of corresponding weights.

Remaining $r''-r'$ elements for $\tilde{X}''$ we take from $\overline{\overline{W}}''$.

Problem of Weighted Threads is just one example of fragmental problems that arise in splitting of optimisation of (0,1) matrices. A series of similar problems arise when different conditions are applied as a consequence of application area modelling. These problems are relatively simple and a large set of them and their solutions are collected in a software library serving the experimentation software system created in this regard.

5. Conclusion

Lagragean relaxation and the related set of techniques is one of the ways of constructing approximation algorithms for hard and unsolved combinatorial problems. A compact class of optimisation problems are effectively modelled in terms of (0,1) matrices. During the Lagrangean relaxation a number of relatively simple optimisation problems arise as a result of splitting the problems into fragmental subproblems. The **Weighted Threads Problem** of this class is solved and the whole chain of approximation is formalised for an example demonstration problem. A software system created on this base provides experimentation environment for treatment of combinatorial NP problems.

Bibliography

1. Nemhauser G. and Wolsey L., Integer and combinatorial optimization, John Wiley & Sons INC., 1999, 763p.
2. Barcucci E., Del Lungo A., Nivat M. and Pinzani R., Reconstructing convex polyominoes from their horizontal and vertical projections, Theoretical Computer Science, 155 (1996) 321-347.
3. Colbourn Charles J., Kocay W.L. and Stinson D.R., Some NP-complete problems for hypergraph degree sequences. Discrete Applied Mathematics 14, p. 239-254 (1986))
4. Aslanyan L., Sahakyan H. Numeral characterization of n-cube subset partitioning, “Optimal Data Structures and Algorithms” conference, 4-6 September, 2006, Rostock, Germany

Authors' Information

Hasmik Sahakyan *– Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am*

Levon Aslanyan *– Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: lasl@sci.am*

TIMED TRANSITION AUTOMATA AS NUMERICAL PLANNING DOMAIN

Alfredo Milani, Silvia Suriani

***Abstract**: A general technique for transforming a timed finite state automaton into an equivalent automated planning domain based on a numerical parameter model is introduced. Timed transition automata have many applications in control systems and agents models; they are used to describe sequential processes, where actions are labelling by automaton transitions subject to temporal constraints. The language of timed words accepted by a timed automaton, the possible sequences of system or agent behaviour, can be described in term of an appropriate planning domain encapsulating the timed actions patterns and constraints. The time words recognition problem is then posed as a planning problem where the goal is to reach a final state by a sequence of actions, which corresponds to the timed symbols labeling the automaton transitions. The transformation is proved to be correct and complete and it is space/time linear on the automaton size. Experimental results shows that the performance of the planning domain obtained by transformation is scalable for real world applications. A major advantage of the planning based approach, beside of the solving the parsing problem, is to represent in a single automated reasoning framework problems of plan recognitions, plan synthesis and plan optimisation.*

***Keywords**: Timed Transition Automata, Automated Planning, Domain*

***ACM Classification Keywords**: F.1.1 Models of Computation I.2.8 Problem Solving, Control Methods, and Search*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

Introduction

Timed transition automata, introduced in [Alur and Dill, 1994], are an extension of finite state automata where the notion of time has been introduced. Transitions take place in specific instants of the time, they can subject to time constraints based on absolute time and/or clocks. Timed automata are very useful to describe the behaviour of systems where the transition or system activities are characterised by prevailing temporal aspects. Many applications of TTA have been developed for control systems and agent models [Ceri et al, 2005]

A main drawback of the automata based model is that they focus on a single aspect of the sequential process, i.e. the problem of recognizing a pattern of actions. On the other hand a planning [Blum and Furst, 1997] based approach to sequential process modeling would allow to manage general issues such as goal attainment problems (i.e. the problem of finding sequence of actions which reach a given state), optimisization problems (sequences of actions which minimise/maximise some given cost function) in a signle framework.

In the following paragraphs it will be shown how the timed transition automata model can be modeled in the framework of numerical parameters planning model, where more general planning and optimisation problems can be posed. Experimental results both for the timed word recognition problem and the general planning problems are also discussed.

Timed Automata

A *Timed Transition Automata* (TTA) [Alur and Dill, 1994] is a finite state machine which is able to recognise timed words, i.e. a sequence of pairs made by symbols over a given alphabet $\sum$ and time values. The pairs in the sequence can be seen as a sequence of logs records, describing user events or system operations annotated with the time in which they occurred. In a TTA it is possible to constrain a certain action to be executed, i.e. a certain transition to occur, only when some time conditions are met. In the following an action-time pair is also referred to as a *token.*

Let us recall more formally some basic concepts related to *Timed Transition Automata*.

Def. *Timed word.* Given a finite alphabet Σ, a timed word on $\sum$, is a finite sequence of pairs or tokens $[(a_0,\tau_0) \ldots (a_k,\tau_k)]$ where $a_i \in \Sigma^*$, $\tau_i \in \Re$ for $i \in [0,k]$ with $\tau_i \leq \tau_{i+1}$ $i \in [0,k-1]$

Def.Timed Language. A *timed language* over an alphabet $\sum$ is a subset of timed words on $\sum$.

Def. Time Transition Automata. A *Timed Transition Automata (TTA)* is a tuple $(\sum, S, s_0, C, E, F)$ where $\sum$ is finite alphabet, S is a finite set of state, $s_0 \in S$ is an initial state, C is a finite set of clocks, $F \subseteq S$ is a set of final acceptance states, $E \subseteq S \times S \times \sum \times 2^C \times \Phi(C)$ defines the transition table for the automata.

Each transition $e \in E$ is a 5-ple $e=<s,s',a,\Lambda,\delta>$ representing a transition from state *s* to state *s'* on input symbol *a* which can occur at a certain time τ when clock constraint δ is verified by the current values of clocks; the transition also resets to 0 the subset $\Lambda \subseteq C$ of clocks.

Clocks are used to express more easily time constraints such as durations relative to sub patterns in the transition diagram. Clocks are usually initialised to 0 and they are updated as time advances.

Given a set X of clocks, the set of clock constraints $\Phi(X)$ includes all the simple constraints conjunctions and negations defined by $\delta := x \leq c \mid c \leq x \mid \neg\delta \mid \delta \wedge \delta$ where $x \in X$ is a clock and *c* is a rational constant.

Def. Run of Timed Transition Automata. A *run* of a timed transition automata records a sequence of legal state transitions and the value of all the clocks when state transitions take place, starting from the initial state s_0.

Def. Timed Language. The language L(A) accepted by an automaton $A=(\sum, S, s_0, C, E, F)$ is the set of all timed words which correspond to consistent runs of the automaton starting with the state s_0 and ending with a final state $s_f \in F$, i.e. a timed word $w=[(a_i,\tau_i)]$ with $i \in [0,k]$ is also $w \in L(A)$ if exists a run from s_0 with each transition $<s,s',a_i,\lambda,\delta>$ taking place at time instant τ_i and the final transition being $<s_{f-1},s_f,a_k,\lambda,\delta>$ for a state $s_f \in F$.

A *domain automaton* can then be defined for representing the legal transitions or, equivalently, the legal sequences of actions which can occur in the system or the agent process to be modelled.

For example, an automaton can be used to describe and recognize the behaviour of a user of an e-learning platform. Assume for instance that the user can perform 7 main operations or activities: *login, lesson, quiz, assignment, chat, view, logout,* and some additional operations: *main menu* which allows to abandon an activity and go back to the main menu; *submit/abandon* which respectively allow to submit the answers of a quiz, or to abandone it without answering. The activities are not all available at the same time, but they are subject to time and precedence constraints. Possible user behaviours are represented by TTAs in the fig.1 and fig,2, with transition labelled by symbols *login,logout,main,clock,submit,chat* and *view* (dashed loops indicate the idle action, i.e. the action of remaining in the current state).

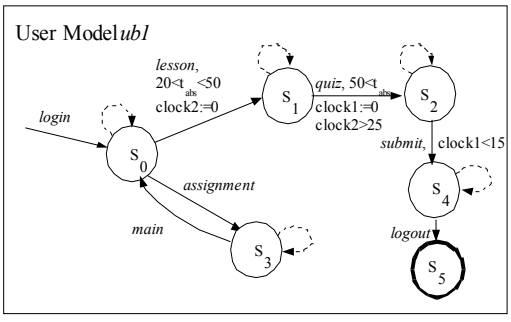


Fig.1 User behavior TTA model *Ub1*

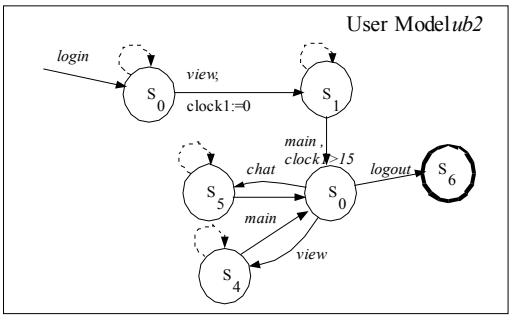


Fig.2 User behavior TTA model *Ub2*

User Model *ub1.* In model *ub1* the user, after entering the e-learning platform (*login* action), can repeat the *assignment* activity many times, but, in order to reach the final state S5, he has to attend the *lesson* until the end for at least 25 minutes (clock2 > 25) and after that he has to *submit* the answer to the quiz.

User Model *ub2.* Behaviour model *ub2*, instead, describes a user which chooses *view* for at least 15 minutes as first activity, and then he/she can alternate *view/chat* without temporal constraints before *logout*.

Let consider, for example, the following timed words, where each element consists of a pair *time stamps* and *symbols.*

```
Seq1:[(login,0),(assignment,10),(main,12),(lesson,22),(main,23),(lesson,24),(quiz,5
1),(submit,65), logout,70)]
Seq2 :  [(login,0),  (view,3),  (main,19),  (chat,20)  (main,25),  (29,chat,29),
(main,35), (view,37), (main,40), (logout,41)]
Seq3: [(logon,0), (view,5), (main,30), (logout,32)]
```

It is easy to see that sequence *Seq1* is an example of user behaviour which is recognised by TTA *ub1*, while sequence *Seq2* and *Seq3* are recognized by *ub2.*

Numerical Parameters Planning Model

In the following we recall some basic notions about the numerical parameters planning model which is used to implement the TTA recognition process. The plan synthesis problem consists in finding a sequence of domain actions which, if executed, transforms a given initial state in a goals state. Planning systems have been widely used to model domain where one or more deliberative actors can modify the state of the world executing a set of predefined available actions. The numerical parameters extension enriches the classical Boolean planning model with the management of numerical resources and goals, moreover effects can depend on numerical continuous parameters of the action instance [Suriani, 2007].The semantics of the model is based on three finite sets: *B*, *N*, and *P*, respectively representing *logical fluents, numerical fluents* and *numerical parameters*. Numerical fluents and numerical parameters are defined in bounded real interval domains.

Definition (State) A *state* is a pair of assignments $s=(s_B, s_N)$ where s_B: B $\rightarrow$ {true, false} assigns truth values to logical fluents, and s_N: N $\rightarrow \Re$ assigns real values to numerical fluents. S_B denotes the set of all possible logical assignments and S_N the set of all possible numerical assignments; finally *S* denotes the set of all possible states.

Definition (Operators) An operator is defined by a triple $o=(X,\pi,\varepsilon)$ where: $X \subseteq$ P are the numerical parameters of *o*; π are the preconditions of *o* and ε are its effects.

Preconditions π are conjunctions of *literals* (i.e. *b* or $\neg b$, where $b \in$ B is a logical fluent) and *numerical constraints* of the form $f_{N\cup X} \otimes 0$, where *f* is a linear function of numerical fluents/parameters and $\otimes \in \{ <, \leq, =, \neq, \geq, > \}$. Effects ε are conjunctions of *literals* and *numerical effects* (i.e. assignments of numerical fluents of the form $u := g_{N\cup X}$ where $u \in$ N, *g* is a linear function of numerical fluents/parameters). Let *O* denote the set of all operators.

Definition (Action Instance) An action instance is defined by a pair (o, σ) where $o=(X,\pi,\varepsilon)$ is an operator and σ a parameter assignment σ: X $\rightarrow \Re$. An action instance (o,σ) is said to be executable in a state $s=(s_B,s_N)$ if logical and numerical conditions hold in *s* and numerical effects are consistent with the domain bounds.

Definition (Action Execution) If an action instance (o,σ) is executable in a state $s=(s_B,s_N)$, the result of its execution is a state $s'=\gamma(s, (o,\sigma)) = (s'_B,s'_N)$, where:

- for each logical fluent $b \in B$ 1) $s'_B(b)$ = true if $b \in \varepsilon$; 2) $s'_B(b)$= false if $\neg b \in \varepsilon$ 3) $s'_B(b) = s_B(b)$ otherwise
- for each numerical fluent $u \in N$ 1) $s'_N(u)=g_{N\cup X}(s_N(u))$ if $u:=g_{N\cup X} \in \varepsilon$ 2) $s'_N(u)=s_N(u)$ otherwise.

Definition (Numerical Parameterized Planning Problem) A numerical parameterized planning problem is a tuple Σ= (B, N, P, S, O, s0, G) where B, N, P, S, O represent boolean fluents, numerical fluents, numerical parameters, states and operators, and

- $s_0 = (s^0_B, s^0_N)$ is the initial state;

- G is a conjunction of literals and numerical constraints defined over $B\cup N$ representing the goal.

Note that goals are defined over (B,N), i.e. goals cannot contain any parameter symbols.

Definition (Solution Plan) A plan, i.e. a sequence of action instances $((o_0,\sigma_0) \ldots, (o_k,\sigma_k))$, is a solution plan for a planning problem $\Sigma = (B, N, P, S, O, s_0, G)$ if the sequence is executable and the goal G holds in the final state.

The sequence of actions is executable when (o_0,σ_0) is executable in s_0 and each action instance (o_i,σ_i) is executable in $s_i = \gamma(s_{i-1}, (o_{i-1},\sigma_{i-1}))$ for each i = 1,…,k. The goal G holds in the final state $s_{k+1} = \gamma(s_k, (o_k,\sigma_k))$, if $\forall g \in G$ when g is a literal g=b (g=¬b) then $s^k_B(b)$=true (s^k_B (b)=false), or when *g* is a numerical constraint $f_{N\cup X}$ then $f_{N\cup X} \otimes 0$ holds in s_{k+1}.

Timed Transition Automata and Equivalent Planning Domain

Since automated planning models encode state transitions, the basic idea of our approach has been to use actions to encode TTA state transitions. The parsing process of a given TTA can be embedded by an appropriate planning domain, where each planning action corresponds to parse a transition in the TTA model, (i.e. corresponds to a legal TTA transition), timed words represent a plan to a final state in the equivalent planning domain.

The current state of TTA is simulated by asserting/negating appropriate fluents. Each planning action representing a TTA transition $<s,s',a_i,\lambda,\delta>$ is executable only if the current simulated state is "s", and if the pair to be parsed is (a_i,τ) where time stamp τ verifies the time constraints δ. The planner can then be used to verify if a timed word, corresponds to a path from the initial TTA state to a final TTA state.

The Planning Domain Problem

Given the TTA $(\sum, S, s_0, C, E, F)$, and given a timed word $w=[(a_i,\tau)]$ it is possible to define a planning domain problem (B, N, P, O, s_0, G) for timed word recognition problem, where:

- B={ *curr_state(s_i), final(s_i), success, curr_tk(l_i), next(l_i, l_j), tk(l_i, a_i ,t_i,d_i)*} is the set of logical fluents where s_i and l_i refer to TTA states and *Tokens*;
- N={ t_{abs} , $t_{\lambda i}$} is the set of numerical fluents;
- P={ t_e } is the set of numerical parameters;
- $O=\{A_{<s,s',a,\lambda,\delta>}, A_{<s,s,a,\lambda,\delta>}, Idle_S , A_f\}\ \forall s\in S, \forall f\in F, \forall_{<s,s',a,\lambda,\delta>} \in E$ is the set of the operators;
- $G= G_B\cup G_\delta$ with G_B={*success*} is the set of literals defined over B and G_δ ={} is the set of numerical constraints defined over N.

Each pair *(a,t)* symbol/time of the timed word is parsed to a 4-pla *(l,a,t,d),* said *token*, where *l* is a sequential identifier, *a* is the symbol encoding the performed action, *t* is the time stamp of the starting time and *d* is the time interval between the action and the next one.

Fluents and TTA States

Given a TTA $(\sum, S, s_0, C, E, F)$ some *logical* and *numerical fluents* are introduced to represent states, tokens, current state, current token and tokens sequence, i.e. timed words.

Logical Fluents.

curr_state(s_i) $\forall s_i \in S$ represents the current state. Note that *curr_state* fluents are used to represent the situation in which the TTA is currently in the state s_i , the domain actions must guarantee that at most one *curr_state(s_i)* can be true at the same time.

tk(l_i, a_i, t_i, d_i) $\forall$ *(l_i, a_i, t_i, d_i)*$\in$*Token* is introduced to represent the token information, l_i is the token id (i.e. sequential identifier), a_i the *action*, t_i the time, d_i is the duration i.e. the time before the next token.

curr_tk(l_i) $\forall$ *(l_i, a_i, t_i, d_i)*$\in$*Token* represents the current token; similarly to *curr_state(s_i),* only one *curr_tk(l_i)* can be true at the same time. The sequential order of the tokens is represented by the fluents *next(l_i,l_j),* where l_i is the successor of l_j in the sequence. A special fluent *curr_tk(init)* represents the initial situation when no tokens are have been parsed yet; conversely, a special fluent *curr_tk(end)* is used to mark the end of the token sequence. Moreover two fluents *next(init,l_1)* and *next(l_k, end)* are also added accordingly.

A set of fluents $final_{Si}$ for each final state $s_i \in F$ and a single logical fluent s*uccess* are also used to specify disjunctive goals.

Numerical Fluents. A numerical fluent t_{abs} is defined to represent the absolute time as it evolves while actions are executed. A numerical fluent t_c is also introduced for each clock $c \in C$.

Initial State

The initial state of the planning problem represents the initial state of the timed automaton and the value of the clocks and of the absolute time are initially set to 0.

$curr\_state(s_0)$= T $curr\_state(s_i) = \perp \ \forall s_i \in S,\ i \neq 0$ s_i is false in I $t_{abs} = 0,\ t_d = 0 \ \forall d \in \delta$

moreover it is also needed to represent the state of the parsing process:

$curr\_tk(init)$ = T $curr\_tk(end) = \perp$ $curr\_tk(l_i) = \perp \ \forall (l_i, a_i, t_i, d_i) \in Token$ $final_{Si}$ = T $\forall s_i \in F\ success = \perp$

the latter two are needed to indicate which are the final states and the fact that the parsing is not yet successful.

TTA transitions and tokens

Appropriate actions $A_{<s,s',a,\lambda,\delta>}$, $A_{<s,s,a,\lambda,\delta>}$, $Idle_{Si}$ and A_{Si} are introduced in the planning domain in order to represent respectively *transitions*, *self-referencing transitions*, *idle states* and the *final disjunctive goal*.

Transitions and Self-referencing Transitions

For each transition $e \in E$, e=<s,s',a, Λ ,δ> of the automata where $s \neq s'$, a planning operator denoted by $A_{<s,s',a,\lambda,\delta>}$ or equivalently by A_e is introduced as follows,

$Pre(A_e)$={ $curr\_state(s) \wedge curr\_tk(l_1) \wedge next(l_1, l_2) \wedge tk(l_1, a, t, d) \wedge \delta \wedge t_{abs} = t$ }

$NumPar(A_e)$={}

$Eff(A_e)$={ $\neg curr\_state(s) \wedge \neg curr\_tk(l_1) \wedge curr\_state(s') \wedge curr\_tk(l_2) \wedge$

$\wedge$ (t_λ:=0, $\forall \lambda \in \Lambda$) ($t_\lambda$:= t_λ+d, $\forall$s.t. $\lambda \in$C and $\lambda \notin \Lambda$) $\wedge$ (t_{abs} := t_{abs}+d)}

where d is the duration of the action whilst l_1 and l_1 are sequential identifiers. The time constraints δ are numerical constraints on the numerical fluents corresponding to the clocks and/or the absolute time; the constraint $t_{abs}=t$ establishes that the transition in TTA takes place at the time t specified by the token.

As shown in figure 3, the basic idea is to contraint the introduced action operator A_e to behave equivalently to transition e:

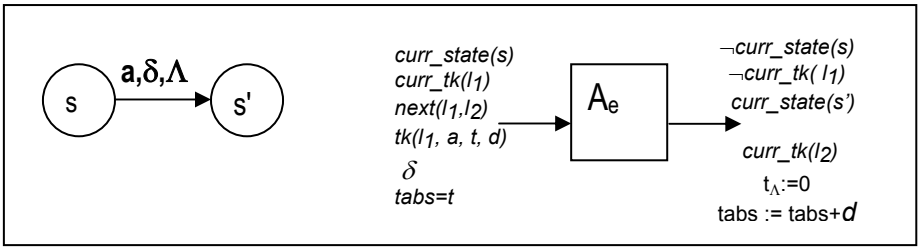


Fig.3 The transition e=<s,s',δ,Λ> and the corresponding action A_e

Also note that the current state and the current token are updated accordingly to the state transition table and to the tokens order, while absolute time is updated with action duration d and clocks are either updated or reset to 0.

A special case is when a transition specifies the same starting and target state, i.e. the corresponding node in the automaton graph contains a self reference loop.

For each transition $e \in E$ of type e=<s, s, a, Λ, δ> it is introduced an action $A_{<s,s,,a,\Lambda,\delta>}$ whose definition differs from the previous one only in the effects, i.e. the negation of current state and the update to the new state, $\neg curr\_state(s) \wedge curr\_state(s)$, are omitted from the action effects since they would lead to inconsistency.

Note that the execution of an action of type $A_{<s,s',a,\Lambda,\delta>}$ or $A_{<s,s,a,\Lambda,\delta>}$ corresponds to parse a token record as required by the precondition *tk(l,a,t,d)*.

Parsing starts from the only one action executable in the initial state, where *curr_state(init)* is true , and it follows the order encoded by the *next* predicates.

Idling state.

If the TTA model admits idling in a state, i.e. remaining in a state while performing no action, then a special *idle operator* $Idle_{Si}$ is added for each state $s_i \in S$ of the TTA in order to model the time flow. The possibility of being idle allows to have gaps in the logs temporal sequence. The idle operators have a quite simple structure since in order to be executed, they do not require either tokens to exists, or time/clock constraints to be verified. On the other hand idle operators contain an additional *numerical parameter* t_e which represents the elapsed time

$Pre(Idle_{Si}) = \{ curr\_state(s_i) \}$

$NumPar(Idle_{Si}) = \{ t_e \}$

$Eff(Idle_{Si}) = \{ \quad (t_{abs} := t_{abs} + t_e) \wedge ((t_\lambda := t_\lambda + t_e, \forall \lambda \in C) \}$

Note that the numerical parameter t_e represents the idling interval and it is used to update the absolute time as well as all the clocks. Numerical parameters are values which are chosen by the planner in order to instantiate the action instance.

TTA Final States & Planning Goals

The TTA recognizes a timed word when it reaches one of the possible final states after parsing all the tokens. These conditions are specified by a disjunctive goals: $curr\_tk(end) \wedge (\vee curr\_state(s_i) \ \forall \ s_i \in F)$

A well known technique [Nebel, 2000] has been used to specify disjunctive goals in a conjunctive planner, a set of dummy actions representing the disjunctive goal is introduce as following:

- for each final state, $\forall s_i \in F$, a dummy operator A_{Si} is added to the set of domain operator O such that:

$Pre(A_{Si}) \equiv \{curr\_state(s_i), final(s_i), curr\_tk(end)\}$, $Eff(A_{Si}) \equiv \{ success \}$

where *success* is a logical fluent representing the end of the user behaviour recognition process.

The fluent *success* will represent the problem goal. *Success* it is true in a state when at least one of the possible action A_{Si} with $s_i \in F$ has been executed, i.e. a final state has been reached (see preconditions $curr\_state(s_i)$, $final(s_i)$) when parsing the last token (precondition *curr_tk(end*)).

Experiments

The TTA planning rules described in the previous paragraph show that the transformation space complexity is linear in the size of the planning domain. On the other hand is not possible to provide a theoretical estimate for plan synthesis time, since it strongly depends on the planner implementation which can employ very efficient strategy expecially for the logical fluents. In order to obtain a general estimate of the effectiveness of the approach we have held systema*tic* experimental tests using PNP (Parametric Numerical Planner), the tests are based on *ub1 and ub2* domains.

PNP has been implemented in C language and performs the graph construction phase and the encoding phase, while the solution of the MILP system is performed by using ILOG CPLEX. The Numerical Parameter Planning model has been implemented using a technique of mixed integer linear programming (MIP) encodings [Wolfman and Weld, 1999]. The algorithm built a planning graph [Kautz and Selman, 1998] with logical fluents and operators ignoring the numerical aspects of the problem, then, the planning graph is encoded as a MIP extended to handle numerical fluents and parameterized action [Vossen et al., 2001, Van de Briel et al., 2005]. A standard MIP solver, ILOG CPLEX is then used to solve the planning problem. The test have been executed on Intel Pentium IV 3.00GHz with 1GB of RAM running the operating system Linux.

The tests has been divided into three classes: positive and negative cases for user behaviour recognition, and planning problems in e-learning domain. Negative cases has been tested for different causes of recognition failure: a) logical failure i.e. action sequences not allowed by the TTA describing the user behaviour and b)numerical failures, action time stamps which violates the numerical time constraint of the TTA. The scalability of the approach has been tested with different users histories, i.e. log sequences of increasing length.

Finally an e-learning planning domain has been modelled to verify the flexibility and expressivity; since the problem does not require to parse any tokens, the fluents of type *tk, curr_tk,*and *next* have been removed from the action descriptions, dummy actions and goals.

Tokens	*Ub1*			*Ub2*		
	Time	Nodes	Var	Time	Nodes	Var
5	0,03	32	174	0,03	34	165
9	0,04	47	345	0,04	49	364
21	0,1	83	1083	0,09	85	1234
33	0,28	119	2253	0,2	121	2536
41	0,49	143	3273	0,34	145	3644
53	1,09	179	5163	0,63	181	5666
61	1,92	203	6663	1,01	205	7254
73	3,56	239	9273	1,74	241	9996
81	5,03	263	11253	2,26	265	12064
93	6,98	299	14583	3,49	301	15526
101	8,23	322	16711	4,55	325	18074

Table 1. Positive Recognition Test for *Ub1*

Tokens	*Ub1*		*Ub1*	
	Log	*Num*	*Log*	*Num*
5	0,03	0,06	0,03	0,09
9	0,03	0,06	0,03	0,09
21	0,03	0,07	0,04	0,11
33	0,06	0,12	0,06	0,18
41	0,09	0,19	0,10	0,29
53	0,18	0,38	0,20	0,58
61	0,26	0,54	0,28	0,82
73	0,47	1,00	0,53	1,53
81	0,68	1,43	0,75	2,18
93	1,11	2,30	1,19	3,49
101	1,50	3,09	1,59	4,68

Table 2. Negative Recognition Test for *Ub1*

The results obtained are completely satisfactory for the three classes of tests. In particular positive user behaviour recognition is quite efficient to be used in real time applications, since the top sequence size of 97 log records are fairly more than the typical user sessions, which consist of less than ten actions, the time performance for twenty actions is worst case not greater than 0.1 seconds. Negative tests on user behaviour recognition were even more efficient than positive tests, in particular it must be noted that negative test of type a), i.e. where the sequence violates a logical constraint, can be detected very efficently in the early plangraph construction phase, and the error detection time is proportional to the length of the correct prefix. Negative tests of type b), i.e. where the timed actions violate the numerical constraints, require the execution of both phases of plangraph construction and LP solving; these tests show an execution time which is slightly minor than the correspondent positive test.

The last class of tests, i.e. e-learning planning problems, is not plotted since the time results are all extremely fast, always below 0.04 seconds for all the posed problems. It would be interesting to investigate in a future extension a task planning approach similar to [Baioletti et al., 1997] where task goals and logical goals can be mixed.

Conclusion

A general method for transforming a *Timed Transition Automata* (TTA) into an equivalent planning domain has been introduced. The main idea of the proposed approach is to built a planning domain model to encode the state transitions of TTA representing behaviours, where each planning action corresponds to parse a time-symbol token, and the sequence of tokens in the time word is given as initial state. The timed word recognition problem is then transformed into the planning problem of finding a parsing plan for the sequence of tokens. The formal TTA to plan transformation is proved to be correct, and it is built in the framework of a numerical parameters planning model, which extends the classical boolean planning models with the management of numerical resources and goals and effects can depend on numerical continuous parameters of the action instance.

One of the relevant advantage in using a planning approach to user task modeling is that behaviour recognition, reachibility and plan optimisation problems can be modeled in a unique framework.

Systematic experiments with PNP, a general purposes parametric numerical planner implementation, show that the approach is effective and scalable for real world application such as user behaviour recognition.

Future works will regard the development of special purpose plan search techniques targeted on the timed word parsing problem, and extending the proposed model with task constraints [Baioletti et al., 1998].

Bibliography

[Alur and Dill, 1994] R. Alur, D. Dill, "A theory of timed automata", *Theoretical Computer Science*, vol. 126, num. 2, p. 183-235, 1994.

[Berendt et al., 2000] B. Berendt, M. Spiliopoulou, "Analysis of navigation behaviour in web sites integrating multiple information systems", *The VLDB Journal*, 9, Springer-Verlag, 2000, pp. 56–75.

[Ceri et al, 2005] S. Ceri, F. Daniel, V. Demaldé, F. M. Facca, "An Approach to User-Behavior-Aware Web Applications", ICWE 5 Proceedings, Sydney, Australia, Springer, 2005

[Masseglia et al., 2005] F. Masseglia, P. Poncelet, M. Teisseire, A. Marascu, "Web Usage Mining: Extracting Unexpected Periods from Web Logs", *TDM 2 - ICDM'05 Proceedings*, Houston, USA, 2005.

[Mühlenbrock, 2005] M. Mühlenbrock, "Automatic Action Analysis in an Interactive Learning Environment", *AIED-2005 Proceedings*, Amsterdam, NL, pp. 73-80.

[Teltzrow et al., 2003] M. Teltzrow, B. Berendt, "Web-Usage-Based Success Metrics for Multi-Channel Businesses", *WebKDD 2003 9th ACM SIGKDD Proceedings*, Washington DC, USA, 2003.

[Baioletti et al., 1998] Marco Baioletti, Stefano Marcugini, Alfredo Milani: Encoding Planning Constraints into Partial Order Planners. *KR98 Proceeding, 6^t Int. Conf. on Principles of Knowledge Representation and Reasoning*,pp.608-616, Morgan Kauffmann 1998, ISBN 1-55860-554-1

[Baioletti et al., 1997] Marco Baioletti Stefano Marcugini, Alfredo Milani: Task Planning and Partial Order Planning: A Domain Transformation Approach. in *Lecture Notes in Computer Science, Vol.1348*, pp.52-63, Springer-Verlag, Berlin, Germany, 1997, ISBN 3-540-64912-8

[Nebel, 2000] Bernhard Nebel: On the Compilability and Expressive Power of Propositional Planning Formalisms. J. Artif. Intell. Res. (JAIR) 12: 271-315 (2000)

[Blum and Furst, 1997] Blum, A., and Frust, M. Fast planning graph analysis. *Artificial Intelligence 90*, 1-2 (1997), 279-298

[Kautz and Selman, 1992] Kautz, H., and Selman, B. Planning as satisfiability. In *10th European Confernce on Artificial Intelligence (ECAI)* (1992), B. Neumann, Ed., Wiley & Sons, pp. 360-363.

[Kautz and Selman, 1998] Kautz, H., and Selman, B. BLACKBOX: A new approach to the application of thorem proving to ploblem solving. In *Working notes of the AIPS-98 Workshop on Planning as Combinatorial Search* (1998), pp. 58-60

[Suriani, 2007] Suriani, S. Numerical Parameters in Automated Planning. *PhD Thesis - Department of Mathematics and Computer Science – University of Perugia.2007.*

[Wolfman and Weld, 1999] Wolfman, S., and Weld, D. The LPSAT engine and its application to resource planning. In *Proc. of IJCAI-99* (!999)

[Vossen et al., 2001] Vossen, T., Ball, M. Lotem, A. and Nau, D. Applying integer programming to AI planning, *Knowledge Engineering Review* 16:85–100, 2001.

[Van de Briel et al., 2005] Van de Briel, M. and Kambhampati, S. Optiplan: Unifying ip-based and graph-based planning. *Journal of Artificial Intelligence Research 24* (2005), 919-931

Authors' Information

Alfredo Milani *–Department of Mathematics and Informatics, University of Perugia, Via Vanvitelli 1, 06100 Perugia, Italy, e-mail: milani@dipmat.unipg.it*

Silvia Suriani *–Department of Mathematics and Informatics, University of Perugia, Via Vanvitelli 1, 06100 Perugia, Italy, e-mail: suriani@dipmat.unipg.it*

P SYSTEMS GÖDELIZATION

Carmen Luengo, Luis Fernández, Fernando Arroyo

***Abstract**: This paper presents a method for assigning natural numbers to Transition P systems based on a Gödelization process. The paper states step by step the way for obtaining Gödel numbers for each one of the fundamental elements of Transition P systems –multisets of objects, evolution rules, priorities relation, membrane structure- until defining the Gödel number of a given Transition P system.*

***Keywords**: Membrane Computing, Transition P System, Gödelization*

***ACM Classification Keywords:** D.1.m Miscellaneous – Natural Computing*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

Introduction

A P system can be defined as a membrane structure in which multiset of objects and evolution rules have been placed in regions defined by membranes. One of the most important characteristic of these computational systems it is the following: objects evolve by the application of evolution rules in a non-deterministic and massively parallel manner all over the system. One way to control the application of evolution rules in membranes is to define a priority relationship among rules in membranes. This priority relation defines a partial order relation and a hierarchy in application of evolution rules inside membranes. Associated to rules, there is one more feature, the capability of dissolving membranes. P systems having these two features (priority relationship and dissolving capability) have been demonstrated that are Turing complete [Păun 2002].

P systems compute transiting from one configuration to the next one by application of evolution rules. A configuration of a P system is defined by the membrane structure of the system and the set of multiset allocated inside membranes of the structure. Hence, a computation is defined as the set of configuration, starting from the initial configuration, the systems transits. It is said that the system performs a successful computation when a configuration in which no one rule can by applied in the system is reaches. This configuration is named "halting configuration". The result of a successful computation is the number of objects present in a determined elemental membrane or the number of objects the system outputs to the environment.

In [Turing 1936] demonstrated that “every computable function corresponds to a Turing machine, and that every Turing machine could be mapped into a unique natural number. As a consequence, the computable functions are enumerable. Moreover, the numbers corresponding to computable functions are known as "Gödel numbers"". Hence, it could be interesting to establish a correspondence between elements of a P system and natural numbers. This gödelization process will permit to obtain some computational benefits. First of all, it is possible to obtain a uniform representation for every element of the P system; it could be possible to reduce the analysis of the system to elementary natural number operations (for example, evolution rules application as divisions [Suzuki 2000]). Secondly this encoding process can produce a new way of packing information and simulating P systems in digital devices with the appropriate algorithms.

This paper proposes a way for obtaining the Gödel number for every element of a P system and some operation for manipulating them with their associated Gödel numbers, and finally the Gödel number of the P system is defined.

Multisets Gödelization

A multiset is defined as a mapping from a non-empty and finite set, *U* over the natural number set. More formally,

$$\begin{aligned} M &: U \to N \\ a &\to M\,a \end{aligned} \tag{1}$$

where *M a* is the number of copies for the *a* element in the multiset *M*. Representing by $\mathcal{M}(U)$ the set of every multiset of objects over the set *U*, it can be defined the following operations:

Multisets operations:

Let M, M_1 y $M_2 \in \mathcal{M}(U)$

- Multisets inclusion:
  - $M_1 \subset M_2 \Leftrightarrow \forall a \in U, M_1\,a < M_2\,a$
- Multisets addition $M_1 + M_2$:
  - $\forall a \in U, (M_1 + M_2)\,a = M_1\,a + M_2\,a$
- Multiset subtraction $M_1 - M_2$:
  - If $M_2 \subset M_1$, $\forall a \in U, (M_1 - M_2)\,a = M_1\,a - M_2\,a$

It can be easily demonstrated that $(\mathcal{M}(U),+)$ is a commutative monoid with identity element.

Gödel number associated to a Multiset

In this section the Gödel number for every multiset $M \in \mathcal{M}(U)$ is defined.

Let $P_m = \{p_1, \ldots, p_m\}$ be the set of the first *m* natural prime numbers starting in 2; and let $U=\{a_1, \ldots, a_m\}$ be a non-empty and finite set of objects with *card(U)=m*, the following one to one map is defined:

$$\begin{aligned} b &: U \to P_m \\ a_i &\to p_i \end{aligned} \tag{2}$$

Moreover, given *b* and P_m there is a function $\mathcal{G}_m$ satisfying:

$$\forall M \in \mathcal{M}(\mathrm{U}), \exists\, \mathrm{n} \in \mathcal{N} \mid \mathcal{G}_m(M)=n \wedge \mathcal{G}_m^{-1}(n) = M$$

defined by:

$$\mathcal{G}_m : \mathcal{M}(U) \to \mathcal{N}$$

$$\begin{aligned} M &\to n = p_1^{M\,a_1} \times \ldots \times p_m^{M\,a_m} \\ \Phi &\to 1 \end{aligned} \tag{3}$$

where $p_i = b(a_i)\ \forall a_i \in U$ as in equation *(2)*.

Definition: $\forall M \in \mathcal{M}(U)$ the *Gödel number* associated to *M* is $\mathcal{G}_m(M)$.

Operations with multiset using Gödel numbers

From the definition of Gödel number associated to a multiset it can be easily demonstrated the following proposition:

Proposition 1: Let M_1 y $M_2 \in \mathcal{M}(U)$ and $\mathcal{G}_m(M_1) = n_1$ y $\mathcal{G}_m(M_2) = n_2$, their associated Gödel numbers, then:

- Multisets inclusion:
  - $M_1 \subset M_2 \Leftrightarrow n_1$ *divides to* n_2

- Multisets addition:
  - $\mathcal{G}_m(M_1 + M_2) = n_1 \times n_2$
- Multisets subtraction: If $M_1 \subset M_2$, *then*
  - $\mathcal{G}_m(M_1 - M_2) = n_1 \div n_2$

From now on, in order to simplify the text, we will represent multiset of objects with small letters. Hence, $u \in \mathcal{M}(U)$ represents one multiset of object in the set U.

Evolution rules Gödelization

In this section the Gödel number for one evolution rule is defined. After that, operations over evolution rules are translated to their associated Gödel numbers. Finally, priority relationship over evolution rules is encoded in Gödel numbers and it is incorporated to the Gödel number associated to the evolution rule.

In order to define evolution rules are needed the following ingredients: a finite set of labels L for numbering membranes in a membrane structure and a non empty and finite set of objects U to define multisets of objects. Let us to represent the set of evolution rules with labels in L and objects in U by $\mathcal{R}(U,L)$.

An evolution rule $r \in \mathcal{R}(U,L)$ is a tuple $r = (u,v,\delta)$ where:

- $u \in \mathcal{M}(U)$
- $v \in \mathcal{M}(U \times \mathcal{D})$ with $\mathcal{D} = \{out, here\} \cup \{in_j \mid j \in L\}$
- $\delta \in \{0,1\}$

Moreover, an evolution rule can be also represented by a set of $(n+2)$ multisets of objects plus one natural number δ representing the rule dissolving capability. Hence,

$$r = (u_a, u_0, u_1, u_2, \ldots, u_n, \delta) \tag{4}$$

where:

- $u_a \in \mathcal{M}(U)$, *the rule antecedent,*
- $u_0 \in \mathcal{M}(U)$, *the multiset to be sent to the environment,*
- $u_i \in \mathcal{M}(U)$, $\forall i \in \{1, \ldots, n\}$, *the multiset to be sent to region i,*
- $\delta \in \{0,1\}$, *the dissolving capability of the rule.*

Using this representation for evolution rules, it is established a gödelization for evolution rules.

Gödel number associated to an evolution rule

First of all, in order to define the Gödel number for an evolution rule $r \in \mathcal{R}(U,L)$ it is necessary to consider a set of $n+3$ natural prime numbers $P = \{p_a, p_o, p_1, \ldots, p_n, p_\delta\}$, and a map g defined as follows:

$$\begin{aligned} g : L \cup \{a,o,\delta\} &\to P \\ a &\to p_a \\ o &\to p_o \\ \delta &\to p_\delta \\ i &\to p_i, \forall i \in \{1,2,\ldots,n\} \end{aligned} \tag{5}$$

Then, from equations *(5)* and *(6)* the Gödel number associated to one evolution rules is defined by:

$$\mathcal{G}_m : \mathcal{R}(U,L) \to \mathcal{N}$$

$$r = (u_a, u_o, u_1, u_2, \ldots, u_n, \delta) \rightarrow p_a^{G_m(u_a)}.p_o^{G_m(u_o)}.\prod_{i=1}^{n} p_i^{G_m(u_i)}.p_\delta^{\delta} \tag{6}$$

Rules operation using Gödel numbers

Let $r_1, r_2 \in \mathcal{K}(U,L)$ be two evolution rules, being $r_1 = (u_1, v_1, \delta_1)$ and $r_2 = (u_2, v_2, \delta_2)$ and let $s \in \mathcal{N}$ be a natural number

It is defined $r_1 + r_2 \in \mathcal{K}(U,L)$ by:

$$r_1 + r_2 = (u_1 + u_2, v_1 + v_2, \delta_1 \vee \delta_2) \tag{7}$$

and $sr_1 \in \mathcal{K}(U,L)$ by:

$$sr_1 = (su_1, sv_1, \delta_1) \tag{8}$$

From *(6)* and *(7)* it can be easily demonstrated that:

$$\begin{gathered} G_m : R(U,L) \rightarrow N \\ r_1 + r_2 \rightarrow p_a^{G_m(u_{1a}+u_{2a})}.p_o^{G_m(u_{1o}+u_{2o})}.\prod_{i=1}^{n} p_i^{G_m(u_{1i}+u_{2i})}.p_\delta^{\delta_1 \vee \delta_2} = \\ p_a^{G_m(u_{1a}).G_m(u_{2a})}.p_o^{G_m(u_{1o}).G_m(u_{2o})}.\prod_{i=1}^{n} p_i^{G_m(u_{1i}).G_m(u_{2i})}.p_\delta^{\delta_1 \vee \delta_2} \end{gathered} \tag{9}$$

and from *(6)* and *(8)*

$$\begin{gathered} G_m : R(U,L) \rightarrow N \\ sr_1 \rightarrow p_a^{(G_m(u_{1a}))^s}.p_o^{(G_m(u_{1o}))^s} \prod_{i=1} p_i^{(G_m(u_{1i}))^s} p_\delta^{\delta} \end{gathered} \tag{10}$$

Rules priority using Gödel numbers

Let $\mathcal{K}_j = \{r_1, r_2, \ldots, r_i\}$ be the set of evolution rules from region *j* of a P system; and let $P_i = \{p_1, p_2, \ldots, p_i\}$ the set of the first *i* prime natural numbers starting in 2.

Then, it is defined the map:

$$\begin{gathered} pri : R_j \rightarrow N \\ r_i \rightarrow p_i \end{gathered} \tag{11}$$

Let ρ_j be the priority relationships associated to $\mathcal{K}_j$, the set of evolution rules in region *j*, and let $ct(\rho_j)$ the transitive and non reflexive closure of ρ_j.

Let $r_k, r_{s1}, r_{s2}, \ldots, r_{st} \in \mathcal{K}_j$ with $(r_k, r_{s1}), (r_k, r_{s2}), \ldots, (r_k, r_{st}) \in ct(\rho_j)$ where rule r_k has a higher priority over rules $r_{s1}, r_{s2}, \ldots, r_{st}$, that is $r_k > r_{s1}, r_k > r_{s2}, \ldots, r_k > r_{st}$ and $r_k \neq r_{s1}, r_k \neq r_{s2}, \ldots, r_k \neq r_{st}$.

It is defined:

$$\begin{gathered} gprior : R_j \rightarrow N \\ r_k \rightarrow pri(r_k).\prod_{i=1}^{t} pri(r_{si}) \end{gathered} \tag{12}$$

Let $r \in \mathcal{K}_j(U,L)$ be an evolution rule in region *j* with label in *L* and objects in *U*. In particular, let $r = (u_a, u_o, u_1, u_2, \ldots, u_n, \delta)$ that rule.

The Gödel number associated to the rule *r* with priority is defined by:

$$G_m: \mathcal{K}_j(U,L) \rightarrow \mathcal{N}$$

$$r = (u_a, u_o, u_1, u_2, \ldots, u_n, \delta) \rightarrow p_p^{gprior(r)} p_a^{G_m(u_a)} . p_o^{G_m(u_o)} . \prod_{i=1}^{n} p_i^{G_m(u_i)} . p_\delta^{\delta} \quad (13)$$

where p_a, p_o, p_i, p_δ are natural prime numbers defined in the one to one mapping g equation *(13)*, and p_p is a different one natural prime number.

Membrane Structure Gödelization

The main element of a P system is the membrane structure. The membrane structure can be represented as a directed and non ordered tree, having the skin membrane as root of the tree. Nodes are membranes and edges represent the relationship to be *'directly included in'*.

Gödel number associated to a membrane tree

Let $P_m = \{p_1, p_2, \ldots, p_m\}$ be the set of the m first natural number starting in *2* and let μ be a membrane structure with labels in $L=\{1,2, \ldots, m\}$.

The Gödel number associated to a membrane is defined by the following map:

$$\begin{aligned} G_m &: \mu \rightarrow P_m \\ & m_k \rightarrow p_k, \quad \forall k \in L \end{aligned} \quad (14)$$

Now, it is possible to define the Gödel number associated to a membrane tree T_μ with labels in L as follows:

$$G_m(T_\mu) = \prod_{k=1}^{m} G_m(m_k), \quad \forall k \in L \quad (15)$$

P systems Gödelization

From the previous Gödel numbers associated to each one of the different components of Transition P systems, it is possible to associate to each Transition P system a Gödel number. Let *Π* be a Transition P system of degree *m*.

$$\Pi = (V, \mu, \varpi_1, \ldots, \varpi_m, (R_1, \rho_1), \ldots (R_m, \rho_{m1}), i_0) \quad (16)$$

where:

- V is a finite and not empty set of objects.
- μ is a membrane structure labelled in a one to one manner from *1* to *m*.
- ϖ_i, $1 \leq i \leq m$, multisets of objects over V associated to regions *1, …, m*.
- R_i, $1 \leq i \leq m$, finite set of evolution rules over V associated to regions *1, …, m.*
- ρ_i, $1 \leq i \leq m$, priority relationships defined over the set of evolution rules R_i.

Gödel number associated to a P system

Now it can be defined the Gödel number associated to the Transition P system *Π* as follows: let $p_1, p_2, \ldots, p_{2m+3}$ be the first $2m+3$ natural prime numbers starting from *2*.

$$G_m(\Pi) = p_1^{Card(V)} p_2^{G_m(T_\mu)} p_3^{G_m(\varpi_1)} \ldots p_{m+2}^{G_m(\varpi_m)} p_{m+3}^{\sum_{r_1^i \in R_1} G_m(r^i{}_{1,}\rho_1)} \ldots p_{2m+2}^{\sum_{r_m^i \in R_1} G_m(r^i{}_{m,}\rho_m)} p_{2m+3}^{i_0} \quad (17)$$

where:

- *Card(V)*, is the cardinal of *V*.

- $G_m(A_\mu)$ is the number defined in *(14)* and *(15)* for the membrane structure μ.
- $G_m(\varpi_i)$, $1 \leq i \leq m$, are the defined numbers in (2) and (3) for the multiset of objects ϖ_i, $1 < i < m$.
- $G_m(ri, \rho_i)$, $1 \leq i \leq m$, is the defined number in *(12)* and *(13)* for evolution rules r_i associated to regions R_i with their corresponding priorities ρ_i, $1 \leq i \leq m$.

Conclusions

Gödel numbers are fundamental in history of computation. Turing in [Turing 1936] showed that every Turing machine could be mapped into a unique natural number and that every computable function corresponds to Turing machine. Hence, the computable functions are enumerable. Here we present the way for mapping Transition P systems into natural number using a Gödelization process. The whole process is described in this paper step by step, in order to define the appropriate Gödel numbers to each one of the different fundamental elements in which Transition P systems are decomposed. Moreover, some operations over multisets of objects and evolution rules have been defined in terms of Gödel numbers associated to them. The different applications of this method to the study of membrane systems are unexplored and it must be subject of study in different aspects related to hardware/software implementations using Gödel numbers, or how to use this encoding process in order to pack information related to P systems and how affect it in the development of algorithms for implementing P systems in digital devices.

Bibliography

[Păun 2002] Gh. Păun, Membrane Computing. An Introduction, Springer-Verlag, Berlin, 2002

[Turing 1936] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2-42, (1936-7), 230-265

[Suzuki 2000] Y. Suzuki, H. Tanaka, On a LISP Implementation of a Class of P Systems, Romanian J. of Information Science and Technology, 3, 2 (2000), 173-186.

[P system Web page] http://psystems.disco.unimib.it/

Authors' Information

Carmen Luengo Velasco – *Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid; Ctra. Valencia, km. 7, 28031 Madrid (Spain);*
e-mail: cluengo@eui.upm.es

Luis Fernández Muñoz – *Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid; Ctra. Valencia, km. 7, 28031 Madrid (Spain);*
e-mail: setillo@eui.upm.es

Fernando Arroyo Montoro - *Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. Valencia, km. 7, 28031 Madrid (Spain);*
e-mail: farroyo@eui.upm.es

FAST LINEAR ALGORITHM FOR ACTIVE RULES APPLICATION IN TRANSITION P SYSTEMS

Francisco Javier Gil, Jorge Tejedor, Luis Fernández

***Abstract**: Transition P systems are computational models based on basic features of biological membranes and the observation of biochemical processes. In these models, membrane contains objects multisets, which evolve according to given evolution rules. In the field of Transition P systems implementation, it has been detected the necessity to determine whichever time are going to take active evolution rules application in membranes. In addition, to have time estimations of rules application makes possible to take important decisions related to the hardware / software architectures design.*

In this paper we propose a new evolution rules application algorithm oriented towards the implementation of Transition P systems. The developed algorithm is sequential and, it has a linear order complexity in the number of evolution rules. Moreover, it obtains the smaller execution times, compared with the preceding algorithms. Therefore the algorithm is very appropriate for the implementation of Transition P systems in sequential devices.

***Keywords:** Natural Computing, Membrane computing, Transition P System, Rules Application Algorithms*

***ACM Classification Keywords:** D.1.m Miscellaneous – Natural Computing*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

Introduction

Membrane computing is a branch of natural computing witch tries to abstract computing models from the structure and the functioning of living cells. The main objective of these investigations consists of developing new computational tools for solving complex, usually conventionally-hard problems. Being more concrete, Transition P systems are introduced by Gheorghe Păun derived from basic features of biological membranes and the observation of biochemical processes [Păun, 1998]. This computing model has become, during last years, an influential framework for developing new ideas and investigations in theoretical computation.

Transition P systems are hierarchical, as the region defined by a membrane may contain other membranes. The basic components of the Transition P systems are the *membranes* that contain chemical elements (*multisets* of objects, usually represented by symbol strings) which are subject to chemical reactions (*evolution rules*) to produce other elements (another multiset). Multisets generated by evolution rules can be moved towards adjacent membranes (parent and children). This multiset transfer feeds back the system so that new multisets of symbols are consumed by further chemical reactions in the membranes.

The P system changes from a configuration to another one making a computation. Each transition or evolution step goes through two sequential steps: application of rules and communication. First, the evolution rules are applied simultaneously to the multiset in each membrane. This process is performed by all membranes at the same time. Then, also simultaneously, all the membranes communicate with their neighbors, transferring symbol multisets.

Most membrane systems are computationally universal: “P systems with simple ingredients (number of membranes, forms and sizes of rules, controls of using the rules) are Turing complete” [Păun, 2005]. This framework is extremely general, flexible, and versatile. Several classes of P systems with an enhanced parallelism are able to solve computationally hard problems (typically, NP complete problems) in a feasible time (polynomial or even linear) by making use of an exponential space.

In this paper we propose a new algorithm for evolution rules application oriented towards the implementation of Transition P systems. The developed algorithm is sequential and, it has a linear order complexity in the number of evolution rules. Moreover, it obtains the smaller execution times, compared with the preceding algorithms. Therefore, due to these characteristics, the algorithm is very appropriate for the implementation of Transition P systems in sequential devices. After this introduction, other related works appear, where the problem that is tried to solve is covered. Next are exposed the formal definitions related to the rules application in Transition P systems. Later the fast linear algorithm for rules application appears developed, finally including the comparison tests and conclusions.

Related Work

In Transition P systems, each evolution step is obtained through two consecutive phases within each membrane: in first stage the evolution rules are applied, and at the second, the communication between membranes is made. This work is centered in the first phase, the application of active rules. It exists several sequential algorithms for rules application in P systems at this moment [Ciobanu, 2002], [Fernández, 2006a] and [Tejedor, 2007], but the obtained results can be improved. In the last mentioned work is introduced an algorithm based on the elimination of active rules: this algorithm is very, interesting because is the first algorithm whose time is only limited by the number of rules, not by the objects multiset cardinality.

Additionally, in [Tejedor, 2006] is proposed a software architecture for attacking the bottleneck communication in P systems denominated “partially parallel evolution with partially parallel communications model” where several membranes are located in each processor, proxies are used to communicate with membranes located in different processors and a policy of access control to the network communications is mandatory. This obtains a certain parallelism yet in the system and an acceptable operation in the communications. In addition, it establishes a set of equations that they allow to determine in the architecture the optimum number of processors needed, the required time to execute an evolution step, the number of membranes to be located in each processor and the conditions to determine when it is best to use the distributed solution or the sequential one. Additionally it concludes that if the maximum application time used by the slowest membrane in applying its rules improves N times, the number of membranes that would be executed in a processor would be multiplied by the square root of N, the number of required processors would be divided by the same factor, and the time required to perform an evolution step would improve approximately with the same factor.

Therefore, to design software architectures it is precise to know the necessary time to execute an evolution step. For that reason, algorithms for evolution rules application that they can be executed in a delimited time are required, independently of the object multiset cardinality inside the membranes. Nevertheless, this information cannot be obtained with most of the algorithms developed until now since its execution time depends on the cardinality of the objects multiset on which the evolution rules are applied.

They have been proposed also parallel solutions -[Fernández, 2006b] and [Gil, 2007]-, but they do not obtain the required performance. The first algorithm is not completely useful, since its run time is not time delimited, and both solutions present efficiency problems due to the competitiveness between the rules, the high number of collisions with the requests and delays due to the synchronization required between processes.

Formal Definitions Related to Rules Application in P Systems

Firstly, this section formally defines the required concepts of objects multisets, evolution rules, evolution rules multiset, and applicability benchmarks (maximal and minimal) of a rule over an objects multiset. Secondly, on the basis of these definitions, requirements are specified for the new algorithm for rule evolution application.

Multisets of Objects

Definition 1: *Multiset of object. Let a finite and not empty set of objects be* O *and the set of natural numbers* N*, is defined as a multiset of object* m *as a mapping*:

$$m : \mathrm{O} \to \mathrm{N}$$
$$o \to n$$

Possible notations for a multiset of objects are:

$$m = \{(o_1, n_1), (o_2, n_2), ..., (o_m, n_m)\}$$
$$m = o_1^{n_1} \cdot o_2^{n_2} \cdot ... \cdot o_m^{n_m}$$

Definition 2: *Set of multisets of objects over a set of objects. Let a finite set of objects be* O*. The set of all the multisets that can be formed over set* O *is defined:*

$$\mathrm{M(O)} = \{m : \mathrm{O} \to \mathrm{N} \mid m \text{ is a } Multiset \text{ over } \mathrm{O}\}$$

Definition 3: *Multiplicity of object in a multiset of objects. Let an object be* $o \in O$ *and a multiset of objects* $m \in M(O)$*. The multiplicity of an object is defined over a multiset of objects such as:*

$$|\ \ |_o :\ \mathrm{O} \times \mathrm{M(O)} \to \mathrm{N}$$
$$(o, m) \to |m|_o = n \mid (o, n) \in m$$

Definition 4*: Weight or Cardinal of a multiset of objects. Let a multiset of objects be* $m \in M(O)$*. The weight or cardinal of a multiset of objects is defined as:*

$$|\ \ | :\ \mathrm{M(O)} \to \mathrm{N}$$
$$m \to |m| = \sum_{\forall o \in O} |m|_o$$

Definition 5: *Multiset support. Let a multiset of objects be* $m \in M(O)$ *and* $P(O)$ *the power set of* O*. The support for this multiset is defined as:*

$$Supp :\ \mathrm{M(O)} \to P(\mathrm{O})$$
$$m \to Supp(m) = \{ o \in \mathrm{O} \ /\ \ |m|_o > 0 \}$$

Definition 6: *Empty multiset. This is the multiset represented by* $\varnothing_{M(O)}$ *and which satisfies:*

$$\varnothing_{\mathrm{M(O)}} \Leftrightarrow |m| = 0 \Leftrightarrow Supp(m) = \varnothing$$

Definition 7: *Inclusion of multisets of objects. Let two multisets of objects be* $m_1, m_2 \in M(O)$*. The inclusion of multisets of objects is defined as:*

$$m_1 \subset m_2 \Leftrightarrow |m_1|_o \le |m_2|_o \quad \forall o \in \mathrm{O}$$

Definition 8*: Sum of multisets of objects. Let two multisets of objects be* $m_1, m_2 \in M(O)$*. The sum of multisets of objects is defined as:*

$$+ :\ \mathrm{M(O)} \times \mathrm{M(O)} \ \to\ \mathrm{M(O)}$$
$$(m_1, m_2) \ \to\ \{(o, |m_1|_o + |m_2|_o)\ \ \forall o \in \mathrm{O}\}$$

Definition 9*: Subtraction of multisets of objects. Let two multisets of objects be* $m_1, m_2 \in M(O)$*, and* $m_2 \subset m_1$*. The subtraction of the multisets of objects is defined as:*

$$- :\ \mathrm{M(O)} \times \mathrm{M(O)} \ \to\ \mathrm{M(O)}$$
$$(m_1, m_2) \ \to\ \{(o, |m_1|_o - |m_2|_o)\ \ \forall o \in \mathrm{O}\}$$

Definition 10: *Intersection of multisets of objects. Let two multisets of objects be* $m_1, m_2 \in M(O)$*. The intersection of multisets of objects is defined as:*

$$\bigcap : \mathrm{M(O)} \times \mathrm{M(O)} \to \mathrm{M(O)}$$
$$(m_1, m_2) \ \to m_1 \bigcap m_2 = \{(o, \min(|m_1|_o, |m_2|_o)\ \ \forall o \in \mathrm{O}\}$$

Definition 11*: Scalar product of multiset of objects by a natural number. Let a multiset be* $m_2 \in M(O)$ *and a natural number* $n \in N$*. The scalar product is defined as:*

$$\cdot : M(O) \times N \to M(O)$$
$$(m, n) \to m \cdot n = \{(o, | m |_o \cdot n) \quad \forall o \in O\}$$

Evolution Rules

Definition 12*: Evolution rule over a set of objects with target in* T *and with no dissolution capacity. Let a set of objects be* O, $a \in M(O)$ *a multiset over* O, $T = \{here, out\} \cup \{in_j / 1 \le j \le p\}$ *a set of targets and* $c \in M(O \times T)$ *a multiset over* $O \times T$*. An evolution rule is defined like a tuple*:

$$r = (a, c)$$

Definition 13: *Set of evolution rules over a set of objects and targets in* T*. This set is defined as:*

$$R(O, T) = \{r \mid r \text{ is a rule over } O \text{ and } T\}$$

Definition 14: *Antecedent of Evolution Rule. Let an evolution rule be* $r \in R(O, T)$*. The antecedent of an evolution rule is defined over a set of objects as:*

$$input : R(O, T) \to M(O)$$
$$(a, c) \to input(r) = a \mid r = (a, c) \in R(O, T)$$

Definition 15: *Evolution rule applicable over a multiset of objects. Let an evolution rule be* $r \in R(O, T)$ *and a multiset of objects* $m \in M(O)$*, it is said that an evolution rule is applicable over a objects multiset if and only if:*

$$\Delta_r(m) \Leftrightarrow input(r) \subset m$$

Definition 16: *Set of evolution rules applicable to a multiset of objects. Let a set of evolution rules be* $R \in P(R(O, T))$ *and a multiset of objects* $m \in M(O)$*. The set of evolution rules applicable to a multiset of objects is defined as:*

$$\Delta^* : P(R(O, T)) \times M(O) \to P(R(O, T))$$
$$(R, m) \to \Delta_R^*(m) = \{r \in R \mid \Delta_R(m) = true\}$$

Property 1: *Maximal applicability benchmark of evolution rule over a multiset of objects*. Let an evolution rule be $r \in R(O, T)$ and a multiset of objects $m \in M(O)$. The maximal applicability benchmark of a rule in a multiset is defined as:

$$\Delta\lceil \ \rceil : R(O, T) \times M(O) \to N$$
$$(r, m) \to \Delta_r\lceil m \rceil = \min\left\{ \frac{| m |_o}{| input(r) |_o} \ \forall o \in Supp(m) \wedge \left| input(r) \right|_o \neq 0 \right\}$$

Property 2: *Minimal applicability benchmark of evolution rule over a multiset of objects and a set of evolution rules*. Let an evolution rule be $r \in R(O, T)$, a multiset of objects $m \in M(O)$ and a set of evolution rules $R \in P(R(O, T))$. The minimal applicability benchmark is defined as the function:

$$\Delta\lfloor \ \rfloor : R(O, T) \times M(O) \times P(R(O, T)) \to N$$
$$(r, m, R) \to \Delta_r\lfloor m \rfloor = \Delta_r\left\lceil m - \left(m \cap \sum_{\forall ri \in R - \{r\}} input(r_i) \cdot \Delta_{ri}\lceil m \rceil \right) \right\rceil$$

Property 3: *An evolution rule* $r \in R(O, T)$ *is applicable to a multiset of objects* $m \in M(O)$ if and only if the maximal applicability benchmark is greater or equal to 1.

$$\Delta_r(m) \Leftrightarrow \Delta_r\lceil m \rceil \ge 1$$

Property 4: The maximal applicability benchmark of a rule $r \in R(O, T)$ *over an object multiset* $m \in M(O)$ *is greater than or equal to the maximal applicability benchmark of the rule in a subset of the object multiset.*

$$\Delta_r\lceil m1 \rceil \geq \Delta_r\lceil m2 \rceil \ \ \forall m1, m2 \in M(O) \ \ | m2 \subset m1$$

Property 5: If the maximal applicability benchmark of a rule $r \in R(O, T)$ over a multiset of objects $m \in M(O)$ is 0, then the maximal applicability benchmark of the rule r over the sum of $input(r)$ and m is equal to the maximal applicability benchmark of the $input(r)$ and equal to 1.

$$\Delta_r\lceil m \rceil = 0 \Rightarrow \Delta_r\lceil input(r) + m \rceil = \Delta_r\lceil input(r) \rceil = 1$$

Multisets of Evolution Rules

Definition 17: *Multiset of evolution rules*. Let a finite and not empty set of evolution rules be $R(O, T)$ and the set of natural numbers N, a multiset of evolution rules is defined as the mapping:

$$M_{R(O,T)} : R(O,T) \rightarrow N$$
$$r \rightarrow n$$

All definitions related to multisets of objects can be extended to multisets of rules.

Definition 18: *Linearization of evolution multiset of rules*. Let a multiset of evolution rules be $m_R = r_1^{k1} \cdot r_2^{k2} \cdot \ldots \cdot r_q^{kq} \in M_{R(O,T)}$ linearization of m_R is defined as:

$$\sum_{i=1}^{q} r_i \cdot k_i \in R(O,T)$$

Requirements of Application of Evolution Rules over Multiset of objects

Application of evolution rules in each membrane of P Systems involves subtracting objects from the objects multiset by using rules antecedents. Rules used are chosen in a non-deterministic manner. The process ends when no rule is applicable. In short, rules application to a multiset of object in a membrane is a process of information transformation with input, output and conditions for making the transformation.

Given an object set $O = \{o_1, o_2, ..., o_m\}$ where $m > 0$, the input to the transformation process is composed of a multiset $\omega \in M(O)$ and $R \in R(O, T)$, where:

$$\omega = o_1^{n_1} \cdot o_2^{n_2} \cdot \ldots \cdot o_m^{n_m}$$
$$R = \{r_1, r_2, ..., r_q\} being\ q > 0$$

In fact, the transformation only needs rules antecedents because this is the part that acts on ω. Let these antecedents be:

$$input(r_1) = o_1^{n_1^i} \cdot o_2^{n_2^i} \cdot \ldots \cdot o_m^{n_m^i} \quad \forall i = \{1, 2, ..., q\}$$

The **output** of the transformation process will be a objects multiset of $\omega' \in M(O)$ together with the multiset of evolution rules applied $\omega_R \in M_{R(O,T)}$.

$$\omega' = o_1^{n'_1} \cdot o_2^{n'_2} \cdot \ldots \cdot o_m^{n'_m}$$
$$\omega_R = r_1^{k_1} \cdot r_2^{k_2} \cdot \ldots \cdot r_q^{k_q}$$

Conditions for making the transformation are defined according to the following requirements:

Requirement 1: The transformation process is described through the following system of equations:

$$n_1 = n_1^1 \cdot k_1 + n_1^2 \cdot k_2 + ... + n_1^q \cdot k_q + n_1^{'}$$
$$n_2 = n_2^1 \cdot k_1 + n_2^2 \cdot k_2 + ... + n_2^q \cdot k_q + n_2^{'}$$
$$...$$
$$n_m = n_m^1 \cdot k_1 + n_m^2 \cdot k_2 + ... + n_m^q \cdot k_q + n_m^{'}$$

That is:

$$\sum_{j=1}^{q} n_i^j \cdot k_j + n_i^{'} = n_i \quad \forall i = \{1, 2, ..., m\}$$

or

$$\sum_{i=1}^{q} input(r_i) \cdot k_i + \omega' = \omega$$

The number of equations in the system is the cardinal of the set O. The number of unknowns in the system is the sum of the cardinals of the set O and the number of rules of R. Thus, the solutions are in this form:

$$(n_1^{'}, n_2^{'}, ..., n_m^{'}, k_1, k_2, ..., k_q) \in \mathrm{N}^{m+q}$$

Meeting the following restrictions:

$$0 \le n_i^{'} \le n_i \quad \forall i = \{1, 2, ..., m\}$$

Moreover, taking into account the maximal and minimal applicability benchmarks of each rule, the solution must satisfy the following system of inequalities:

$$\Delta_{r_j} \lfloor \omega \rfloor \le k_j \le \Delta_{r_j} \lceil \omega \rceil \quad \forall j = \{1, 2, ..., q\}$$

Requirement 2: No rule of the set R can be applied over the multiset of objects ω', that is:

$$\Delta_r(\omega') = false \quad \forall r \in \mathrm{R}$$

Having established the above requirements, the system of equations may be incompatible (no rule can be applied) determinate compatible (there is a single multiset of rules as the solution to the problem) or indeterminate compatible (there are many solutions). In the last case, the rule application algorithm must provide a solution that is randomly selected from all possible solutions in order to guarantee non-determinism inherent to P systems.

Fast Linear Algorithm for Active Rules Application in Transition P Systems

This section describes the fast linear algorithm for active rules application to a multiset of objects whose execution time depends on the number of rules. The initial input is a set of active evolution rules for the corresponding membrane -the rules are applicable and useful- and the initial membrane multiset of objects. The final results are the complete multiset of applied evolution rules and the obtained multiset of objects after rules application.

The algorithm is based on the one by one elimination of rules: when a rule has been applied to its maximal applicability benchmark, this rule lets be active, and therefore it is eliminated. The algorithm finishes when all rules have been eliminated. The algorithm is made up of two phases:

1. At the first phase all rules belonging to the set of active rules -except one- are applied a random number of times between 0 and its maximal applicability benchmark. In this way, each active rule has a possibility of being applied.

2. In the second phase, all the rules -beginning by the one excluded of the previous phase- are applied to its maximal applicability benchmark. Consequently, there it is not left any rule applicable, and the algorithm finishes generating like result the multiset of rules applied and the final multiset of objects.

In order to facilitate the explanation of the algorithm, the set of initially active rules is represented like an ordered sequence R and an auxiliary structure called Active. The position of any rule r_i in the sequence is i. Active[i] indicates if the rule r_i continues active. The rule excluded in first stage is the one that is in the last position of the sequence (it can be any rule of the set of active rules). The pseudocode of the algorithm is as follows:

```
( 1)  ω' ← ω
( 2)  ωR ← ØMR(U)
( 3)  FOR i = 1 TO |R| - 1 DO // Phase 1
( 4)      BEGIN
( 5)      Max ← ΔR[i](ω')
( 6)      IF ( Max ≠ 0 ) THEN
( 7)          BEGIN
( 8)          K ← random(0, Max)
( 9)          ωR ← ωR + {R[i]^K}
(10)          ω' ← ω' − input(R[i]) · K
(11)          Active[i] = (K < Max)
(12)          END
(13)      ELSE Active[i] = false
(14)      END
(15)
(16)  Active[|R|] = true
(17)  FOR i = |R| DOWNTO 1 DO // Phase 2
(18)      IF ( Active[i] ) THEN
(19)          BEGIN
(20)          Max ← ΔR[i](ω')
(21)          ωR ← ωR + {R[i]^Max}
(22)          ω' ← ω' − input(R[i]) · Max
(23)          END
```

As it has been previously indicated, the algorithm is made up of two phases. In first stage is offered the possibility to all the rules -except one- to be applied between 0 and their maximum applicability benchmark. In addition is determined if a rule lets be active. Rules can let be active in this stage due to two possible reasons: a) the rule has been applied to its maximum applicability, or b) other preceding rules have consumed the necessary objects so that the rule can be applied.

The second phase begins supposing like active the last rule (observe that this is not necessarily certain). Next, beginning by the one excluded of the first phase, all the supposedly active rules are applied to its maximum applicability. After this step all the rules let be active, and the application algorithm finished their execution. As it can be seen, the algorithm executes a finite and well-known number of operations, which only depends on the initial number of active rules.

In the next sections we are going to demonstrate the correctness of the exposed algorithm, as well as the efficiency analysis.

Algorithm Correctness

The presented algorithm is correct because:

Lemma 1: *The algorithm is finite.*

Proof: The first two lines are basic operations. The first loop -from line (3) to (14)- is exactly executed $|\mathrm{R}| - 1$ times, and its body only contains simple operations. The second loop -from line (16) to (23)- is exactly executed $|\mathrm{R}|$ times, and also its body only contains simple operations.

Lemma 2: *No evolution rule is applicable to ω'.*

Proof: The sequence R initially contains all rules applicable to *ω'*. Owing to **property 3** we know that a rule with a maximal applicability benchmark equal to zero is not applicable. After the execution of the second phase of the algorithm, the maximal applicability of all the rules is zero. Therefore, at the end of the algorithm execution, it is not left any rule applicable to *ω'*.

Lemma 3: *Any result generated is a possible solution.*

Proof: The multiset of rules applied ω_R is obtained by the multiple applications of the active rules in both phases. In addition, since in the second phase each active rule is applied to its maximal applicability benchmark, after the execution of the algorithm no rule is applicable over *ω'* (requirement 2), and the result generated is a possible solution.

Lemma 4: *Any solution possible is generated by the algorithm*

Proof: Phase 1 of the algorithm -from line (3) to (14)- guarantees that any possible solution can be generated. It is enough whereupon the appropriate number is generated in line 8, when the number of applications of a rule is determined. In the second phase it would be only needed to apply the last rule the appropriate number of times.

Lemma 5: *The algorithm is not determinist*

Proof: This occurs when a rule is not the last one in the set, it is applied a randomly determined number of times (sentence 8) between zero and its maximal applicability value.

Efficiency Analysis

Examining the algorithm it is possible to observe that in the two phases, the heaviest operations are those for calculating the maximal applicability benchmark (sentences 5 and 20), the scalar product of the *input* of a rule by a whole number and the difference of two multisets (sentences 10 and 22). These operations are made in both phases in the worse case. All these operations are linearly dependant on the cardinal of the *multiset support* ω.

$$\#operations\_per\_iteration \approx 3 \cdot Supp(\omega)$$

Moreover, the worst case of the fast linear algorithm occurs when sentences 6 and 11 are evaluated always affirmatively, or what is the same, when no rule becomes inactive after the execution of first stage. In this case, there is no improvement in the behavior of the algorithm and the number of iterations executed is:

$$\#iterations = (|\mathrm{R}| - 1) + |\mathrm{R}| = 2 \cdot |\mathrm{R}| - 1$$

Therefore, the number of operations executed at worst case by the algorithm is:

$$\#operations = (2 \cdot |\mathrm{R}| - 1) \cdot 3 \cdot Supp(\omega)$$

So the execution time of the algorithm at worst is linear dependant of the number of rules.

Comparison Tests

The experimental tests have compared the execution time of the *Fast Linear algorithm* (FLA) with the one that was fastest until now, that is *Active Rules Elimination* (ARE) algorithm [Tejedor, 2007]. The experimental trial game used to test both algorithms has taken into account 3 parameters:

1. *Number of objects of the multiset*. In comparative the value of this parameter is 16
2. *Number of rules (q)*. The value of q has taken all the values from the set {1, 2, 4, 8, 16, 32, 64, 128}
3. *Relationship between the cardinal of the multiset and the cardinal of the sum of inputs of the active rules set (r).* The value of r has taken all the values from the set {1, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7}

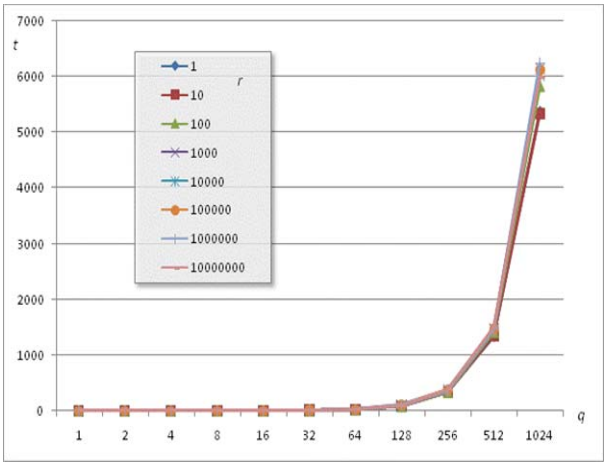


Figure 1.- Evolution of the execution times difference

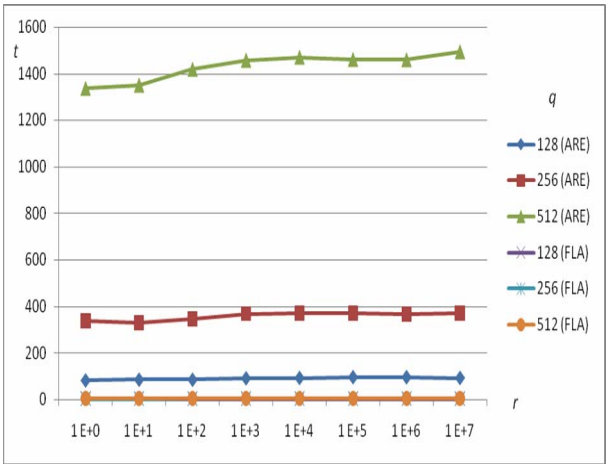


Figure 2.- Execution time of ARE and FLA

Figure 1 shows a graphic with the evolution of the execution times difference obtained in the tests between the REA and the FLA. Each curve of the graphic represents the difference of execution time of the ARE algorithm with regards to the execution time of FLA algorithm for each of the values of the relationship of the cardinals (r). In this graphic you can see that FLA algorithm is always better than ARE algorithm independently of the number of rules and the relationship between cardinals.

In Figure 2 it can be observed that the execution time grows modestly -with both algorithms- with the value of r. The parameter that influences more in the results is the number of rules (q). The obtained results are logical considering that the complexity of the ARE algorithm is order of square of q, and the complexity of the FLA is q linear.

Conclusions

This paper introduces a new algorithm for active rules application to a multiset of objects based on rules elimination in transition P systems. This algorithm attains a certain degree of parallelism, as a rule can be applied a great number of times in a single step. The number of operations executed by the algorithm is time delimited, because it only depends on the number of rules of the membrane. The number of rules of the membrane is well known static information studying the P system, thus allowing determining beforehand the algorithm execution time. This information is essential to calculate the number of membranes that have to be located in each processor in distributed implementation architectures of P systems to achieve optimal times with minimal resources.

We think that the presented algorithm can represent an important contribution in particular for the problem of the application of rules in membranes, because it presents high productivity and it allows estimate the necessary time to execute an evolution step. Additionally, this last one allows making important decisions related to the implementation of P systems, like the related ones to the software architecture.

Bibliography

[Ciobanu, 2002] G. Ciobanu, D. Paraschiv, "*Membrane Software. A P System Simulator*". Pre-Proceedings of Workshop on Membrane Computing, Curtea de Arges, Romania, August 2001, Technical Report 17/01 of Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, Spain, 2001, 45-50 and Fundamenta Informaticae, vol 49, 1-3, 61-66, 2002.

[Ciobanu, 2006] G. Ciobanu, M. Pérez-Jiménez, Gh. Păun, "*Applications of Membrane Computing*". Natural Computing Series, Springer Verlag, October 2006.

[Fernández, 2006a] Fernández, L. Arroyo, F. Castellanos, J. et al (2006) "*New Algorithms for Application of Evolution Rules based on Applicability Benchmarks*". BIOCOMP 06, Las Vegas (USA)

[Fernández, 2006b] L. Fernández, F. Arroyo, J. Tejedor, J. Castellanos. "*Massively Parallel Algorithm for Evolution Rules Application in Transition P System*". Seventh Workshop on Membrane Computing, WMC7, Leiden (The Netherlands). July, 2006

[Gil, 2007] Gil, F. J. Fernández, L. Arroyo, F. et al "*Delimited Massively Parallel Algorithm based on Rules Elimination for Application of Active Rules in Transition P Systems*" i.TECH-2007. Varna (Bulgaria).

[Păun, 1998] G. Păun. "*Computing with Membranes*". In: Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Păun, 2005] G. Păun. "*Membrane computing. Basic ideas, results, applications*". In: Pre-Proceedings of First International Workshop on Theory and Application of P Systems, Timisoara (Romania), pp. 1-8, September, 2005.

[Tejedor, 2006] J. Tejedor, L. Fernández, F. Arroyo, G. Bravo. "*An Architecture for Attacking the Bottleneck Communications in P systems*". In: Artificial Life and Robotics (AROB 07). Beppu (Japan), January 2007.

[Tejedor, 2007] J. Tejedor, L. Fernández, F. Arroyo, A. Gutiérrez. "*Algorithm of Active Rules Elimination for Evolution Rules Application*". In 8th WSEAS Int. Conf. on Automation and Information, Vancouver (Canada), June 2007.

Authors' Information

F. Javier Gil Rubio *– Dpto. de Organización y Estructura de la Información, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: jgil@eui.upm.es*

Jorge A. Tejedor Cerbel *- Dpto. de Organización y Estructura de la Información, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: jtejedor@eui.upm.es*

Luis Fernández Muñoz *- Dpto. de Lenguajes, Proyectos y Sistemas Informáticos, E.U. de Informática. Natural Computing Group, Universidad Politécnica de Madrid, Spain; e-mail: setillo@eui.upm.es*

EVALUATION OF PARETO/D/1/K QUEUE BY SIMULATION

Seferin Mirtchev, Rossitza Goleva

***Abstract**: The finding that Pareto distributions are adequate to model Internet packet interarrival times has motivated the proposal of methods to evaluate steady-state performance measures of Pareto/D/1/k queues. Some limited analytical derivation for queue models has been proposed in the literature, but their solutions are often of a great mathematical challenge. To overcome such limitations, simulation tools that can deal with general queueing system must be developed. Despite certain limitations, simulation algorithms provide a mechanism to obtain insight and good numerical approximation to parameters of queues. In this work, we give an overview of some of these methods and compare them with our simulation approach, which are suited to solve queues with Generalized-Pareto interarrival time distributions. The paper discusses the properties and use of the Pareto distribution. We propose a real time trace simulation model for estimating the steady-state probability showing the tail-raising effect, loss probability, delay of the Pareto/D/1/k queue and make a comparison with M/D/1/k. The background on Internet traffic will help to do the evaluation correctly. This model can be used to study the long-tailed queueing systems. We close the paper with some general comments and offer thoughts about future work.*

***Keywords**: Pareto distribution, delay system, queueing analyses, simulation model, peak traffic modelling;*

***ACM Classification Keywords:** G.3 Probability and statistics: queueing theory, I.6.5 Model development*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

Introduction

Managed IP networks have become a dominant factor in bringing information to users on a worldwide basis. Until recently, IP networks supported only a best effort service. This limitation has not been a problem for traditional Internet applications like web and email, but it does not satisfy the needs of many new applications like audio and video streaming, which demand high data throughput capacity (bandwidth) and have low-latency requirements. Thus, it is becoming increasingly important to provide Quality of Service (QoS) in managed IP networks.

As pointed out by several authors who have been collecting traffic data from the Internet, there is no a queueing theory method for queue analyses when one is given a set of packet interarrival times. Obviously, one could fit the resulting data to a distribution and then use a queueing model if it exists. There are some papers concerning batch arrivals like [Khadjiivanov, 1993]. Traffic growth and its influence to the congestion management is demonstrated in [Tsankov, 2007]. Internet traffic can be described as having one or more of the following related characteristics [Cao, 2004, Salvador, 2004]: Self-similar (or fractal) traffic traces; Long-range dependence; Burstiness on multiple scales; Long- or heavy-tailed packet interarrival times or service requirements.

There has been a substantial amount of literature on analyzing and characterizing the traffic appearing on the Internet. The Internet traffic data are well known to possess extreme variability and bursty structure in a wide range of time scales. This characteristic is not found on the Poisson process. The properties can be characterized by self-similar process. The large variation pertaining to the self-similar nature of data traffic causes congestion problems in the data network. The arrival process with Pareto distributed interarrival time is a popular model of self-similar processes.

The queue performance of Pareto/M/1/k was studied by simulations in [Koh, 2003]. They are investigated the queue behaviour with Pareto interarrival distribution. By numerical analysis and simulations, they have been analyzed the asymptotic and the exact loss probabilities of GI/M/1/k to show the big discrepancy between the asymptotic and the actual loss probability and propose a model for the loss probability of Pareto/M/1/k as a function of the buffer size and the geometric parameter.

The Pareto distribution is a model for nonnegative data with a power law probability tail. A natural upper bound truncates the probability tail in many practical applications. An estimators are derived for the truncated Pareto distribution in [Inmaculada, 2006]. They investigate distribution properties and illustrate its applicability in practice.

The simulation of systems using heavy-tailed distributions presents difficulties and needs efficient methods to study. In [Argibay, 2003] there is a trial to go into insight nature of simulation difficulties of M/G/n queues with G heavy-tailed distribution. They have proposed and developed a method to speed up simulations and used M/G/1 systems as workbenches since they have some analytical results to check the results.

Stochastic simulation has become a well established paradigm used in performance evaluation of various complex dynamic systems. In [Eickhoff, 2006] a method for estimating time evolution of several quantiles within some time interval is described. It is based on independent replications and its capability is demonstrated by simulating processes with different kinds of stationary, non-stationary or transient behaviour.

The concept of self-similarity (or fractal behaviour) is the best understood by looking at [Fernandes, 2003]. The use of synthetic self-similar traffic in computer networks simulation is of vital importance for the capturing and reproducing of actual Internet data traffic behaviour. Fernandes uses a technique for self-similar traffic generation that is achieved by aggregating On/Off sources where the active (On) and idle (Off) periods exhibit heavy tailed distributions. This work analyzes the balance between accuracy and computational efficiency in generating self-similar traffic and presents important results that can be useful to parameterize existing heavy tailed distributions such as Pareto, Weibull and Lognormal in a simulation analysis.

The Pareto distribution is a special heavy tailed distribution called a power-tailed distribution. It is found to serve as adequate model for many situations. Gross and al. [Gross, 2003] investigated many difficulties in simulating queues with Pareto service. They considered truncated Pareto service.

A method for studying Pareto queues is presented in [Fischer, 1999]. The paper discusses the properties and use of the Pareto distribution. The method is used to study the Pareto/M/1 queue and look at the M/Pareto/1 queue. The first could be used to model arrivals of packets at a packet switched network, and the second, the time to transmit files through such a network.

The Pareto distribution has various forms. A one and two-parameter form is considered in [Fischer, 2005]. The two Pareto forms are studied in detail. It is shown that the usage of the two-parameter Pareto results in lower congestion than the comparable one-parameter Pareto.

Some limited analytical derivation for queueing models whit Pareto distribution is proposed in the literature, but their solutions are often of a great mathematical challenge. To overcome such limitations, simulation tools that can deal with general queueing systems have to be developed. Despite certain limitations, simulation algorithms provide a mechanism to obtain insight and good numerical approximation to parameters of networks of queues.

This paper presents a stochastic simulation method for studying Pareto queues. The paper discusses the properties and use of the Pareto distribution. We make the comparison between Pareto/D/1/K and M/D/1/K and propose a real time trace simulation model for estimating the steady-state probability showing the tail-raising effect, the loss probability and delay. The background on Internet traffic will help to do the evaluation correctly. This model can be used to study the long-tailed queueing systems.

Generalised Pareto distribution

The most common choice for telecommunication network design is based on the exponential assumption. Usual choice is the Poisson arrival of the calls or sessions and exponential holding times. However, networks and applications of today generate a traffic that is bursty over a wide range of time scales. A number of empirical studies have shown that the network traffic is self-similar or fractal in nature.

The Pareto distribution, named after the Italian economist Vilfredo Pareto, is a power law probability distribution that coincides with social, scientific, geophysical, actuarial, and many other types of observable phenomena.

The family of Generalized Pareto Distributions (GPD) has three parameters: the location parameter μ, the scale parameter σ and the shape parameter ξ.

The cumulative distribution function of the GPD is:

$$F(x) = 1 - (1 + \frac{\xi(x-\mu)}{\sigma})^{-1/\xi} \quad . \tag{1}$$

We choose these substitutions

$$\eta_0 = \frac{\xi}{\sigma}; \quad \lambda = \frac{\sigma}{\sigma^2 - \xi}; \quad \mu = 0 \quad . \tag{2}$$

Therefore, we receive another form of the generalized-Pareto distribution:

$$F(t) = 1 - (1 + \eta_o t)^{-\left(1 + \lambda/\eta_o\right)} \tag{3}$$

The mean value of the generalized-Pareto distribution is:

$$m_o = 1/\lambda \tag{4}$$

The mean value is the average interarrival time for our study. The parameter λ is the call arrival intensity.

The variance of the Generalized Pareto Distribution is:

$$d_i = \frac{\lambda + \eta_o}{\lambda^2(\lambda - \eta_o)}, \quad 0 \le \eta_o \le \lambda \tag{5}$$

It follows that the probability density function of the GPD is:

$$f(t) = (\eta_o + \lambda)(1 + \eta_o t)^{-\left(2 + \lambda/\eta_0\right)} \tag{6}$$

It is convenient to define the mean value and variance of the arrival stream. We can easy calculate the parameter η_o (the ratio of the shape and scale parameter):

$$\eta_o = \lambda\left(1 - \frac{2}{d_i\lambda^2 + 1}\right) \tag{7}$$

Random number generation

Many programming languages do not yet recognize the Pareto distribution. In the field of telecommunications, the Pareto distribution is widely used to estimate the interarrival and service times.

One can easily generate a random sample from Pareto distribution by using inverse distribution function. Given a random variable U with uniform distribution on the unit interval (0;1), the random variable *x* is Pareto-distributed.

$$x = \frac{U^{-\frac{\eta_o}{\eta_o + \lambda}} - 1}{\eta_o} \tag{8}$$

Uniformly distributed pseudo-random numbers in the space (0,1] are usually referred to as *random numbers*, whereas random numbers following any other distribution are referred to as *random variates* or *stochastic variates.*

Pareto/D/1/k Simulation Model Description

Recall that in standard queueing notation, A/B/C, "A" represents the arrival distribution, "B" the service distribution, and "C" the number of servers. "M" means "memoryless", which in this context implies Poisson distribution for arrival rates and exponential distribution for service times. Our simulation model is used to study the Pareto/D/1/k queue. It could be used to model arrivals of packets at a packet switched network.

Let us consider a single server queue Pareto/D/1/k with a Pareto input stream, which is defined by arrival intensity λ, variance of the interarival time d_i, constant service time τ and limited waiting room k. This queueing system with peak input stream and constant service time is a non-Markovian model (Figure 1). It is assumed that customers are served in FCFS order.

Simulations are the main tools for studying the performance of telecommunication networks and we will analyse Pareto/D/1/k queue using simulation.

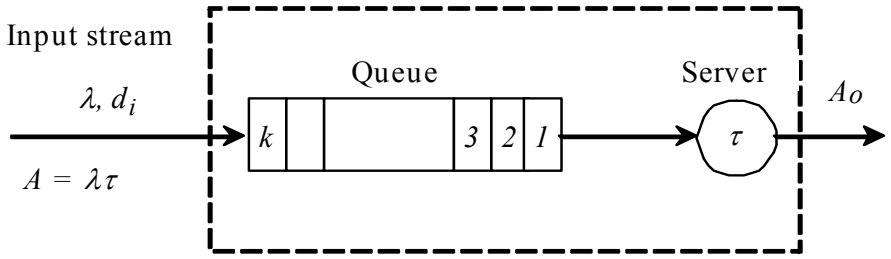


Figure 1. *Pareto input stream model with constant service time and finite waiting positions.*

The arrival process in Pareto/D/1/k queue is considered to be a renewal process. G/D/1/k queues, where G means a heavy-tailed distribution of interarrival time, are used to model queue systems where a range of values of the interarrival time, whose probability is very low, have a drastic impact on the overall performance of the system. The Pareto distribution is one of these heavy-tailed distributions and it is proposed to describe peak streams in the packet switched networks. The accurate analytical treatment of Pareto/D/1/k systems is very difficult and in many cases it cannot be applied. Simulation is a possible method to study. Simulations with heavy-tailed random variables present some additional difficulties. A care must be taken during analyses of the results of these simulations. It is necessary to have accurate and efficient simulation methods. The efficacy is important because we need to generate big quantities of data for our simulation study and be accurate enough. The data accuracy can be estimated by means of comparisons with known results from simpler systems with analytical solution. One of these simpler queue systems that is studied analytically is the M/D/1/k queue. This queue is used as a workbench for more efficient simulation methods, able to deal with the heavy-tail difficulties.

We develop a real time trace simulation algorithm for evaluating the state probabilities of the queueing system, the call congestion probability and the mean time in the queue. We use batch mean method for output results analysis and choose a confidence probability 95%. We define 20 batches and generate 20000 calls in every batch. We introduce an initial bias to eliminate the influence of the transient behaviour and time intervals between batches to received independent estimates of the call congestion probability and the mean queueing time. We describe the accuracy of the estimates by means of a confidence interval, which with a given probability (95%) specifies how the estimate is placed relatively to the unknown theoretical value, using the Student's t-distribution with 19 degrees of freedom. This organization of our algorithm leads to good accuracy from a practical point of view. The relative errors of the presented results are less than 10%.

Random errors are caused by the stochastic variations of the simulation. They appear because every simulation is similar to a statistical experiment. The next source of error is the bias of the estimator itself, being often called the systematic error. This kind of error usually appears if assumptions about the analyzed data are true only approximately or asymptotically. If both the variance and the bias tend to zero for large number of observations the estimator is called consistent.

Pareto/D/1/k System Performance Measures

In this section, we present some of the parameters used in our analysis and the results of the simulation runs. We will illustrate these parameters by running a different scenario.

OFFERED TRAFFIC

The offered traffic *A* is calculated by means of the average arrival rate and the constant service time

$$A = \lambda\tau \quad . \tag{9}$$

BLOCKING PROBABILITY

The call congestion probability *B* is defined and evaluated by the simulation program as the ratio of lost and arrival calls. It can be calculated by offered and carried traffic

$$B = \left(A - A_o\right)/A \quad . \tag{10}$$

MEAN QUEUEING TIME

The real time simulation gives as possibility to calculate the queueing time for every arrival call and it is easy to obtain the mean queueing time.

Simulation Results

In this section, we give numerical results obtained by a Pascal program on a personal computer. The described models are tested on a computer over a wide range of arguments.

Figure 2 illustrates the stationary probability distribution in a single server queue Pareto/D/1/k (P/D/1/k on the Figures below) with a Pareto input stream, 0.85 erl offered traffic, 30 waiting positions and different variance of the interarrival time. It is seen that when the variance increases the probability that the queue is full increases significantly.

Figure 3 presents the call congestion probability in a single delay system with 30 waiting positions, different offered traffic and different variance of the interarrival time. When the offered traffic is comparatively small (0.8 erl) the influence of the variance of the call congestion probability is great.

Figure 4 shows the queueing time as function of the offered traffic when the number of queueing positions is 30, the service time is 1 second and different variance of the interarrival time.

It is shown that the influence of the variance of the input stream over the performance measures is significant. The heavy-tailed condition decisively contributes to raise the congestion and waiting time.

The computer simulation of Pareto/D/1/k queues presents important difficulties due to the slow decaying tail of the Pareto distribution. This makes extremely high values, with great influence on the statistical figures of the system. The probabilities are so low that in case we want to simulate the physical underlying processes, generating demanded times and time arrivals, the cost in time will probably be prohibitive if we want accurate results. This forces to use all our knowledge of the statistics of the system inner processes, so the simulation can noticeably speed up.

Conclusion

We have presented a simulation method for evaluating the Pareto/D/1/k queueing systems. We have demonstrated its use by presenting numerical results. These results have shown that the Pareto distribution change significantly the queue behaviour. In the case of Pareto/D/1/k, congestion occurred even when the load is sufficiently small. But for that queue, the long-tailed nature of the Pareto helps to clear out congestion when a large interarrival time occurred. Our model can be applied for all Pareto/D/1/k systems independently of the value of the parameters.

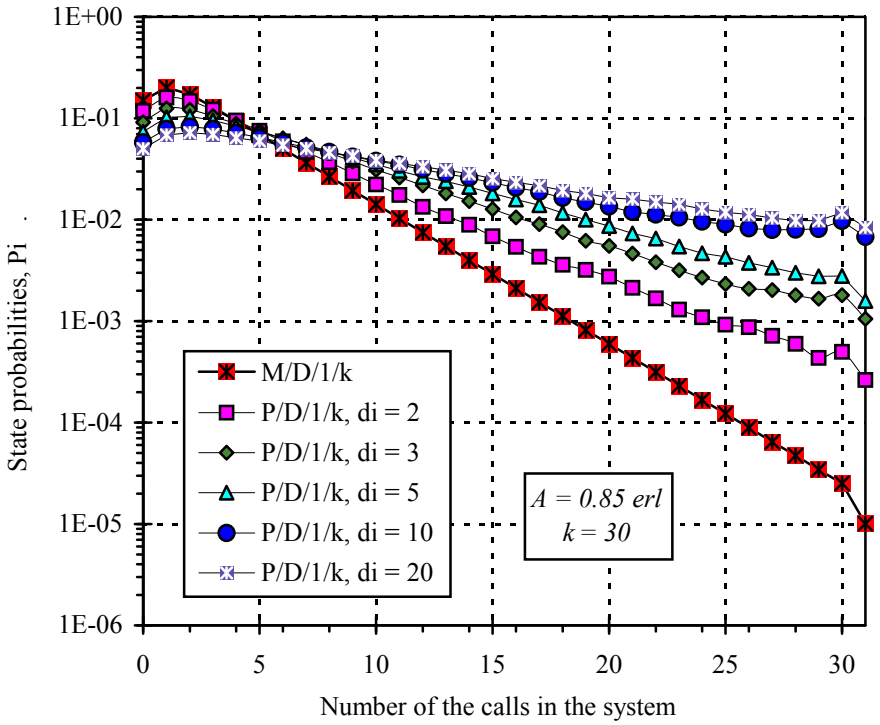


Figure 2. *Stationary probability distribution of the Pareto/D/1/k when the offered traffic is A = 0.85 erl,* the number of the waiting rooms *k = 30* and different peakedness of the input stream

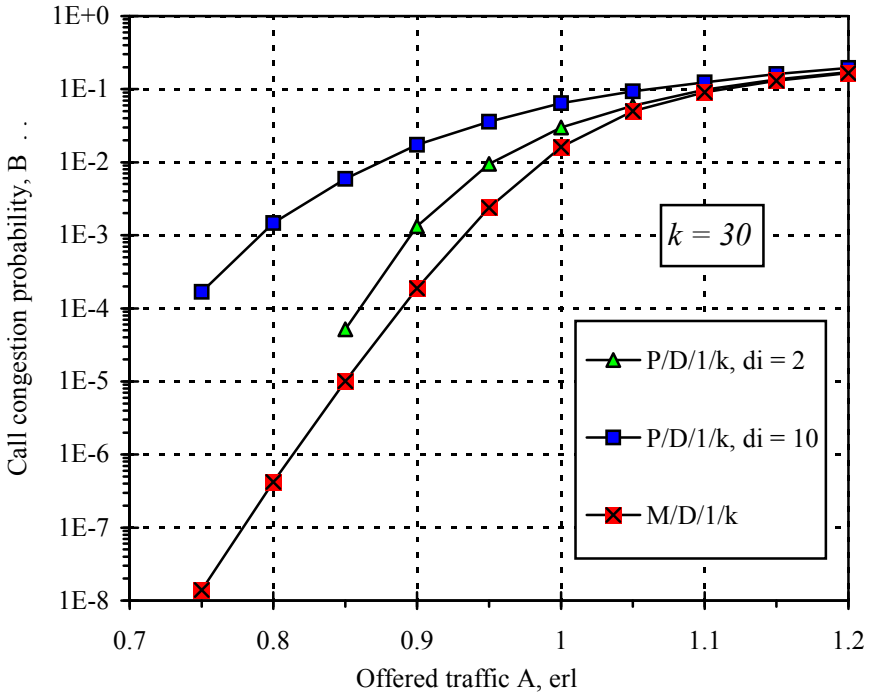


Figure 3. *Call congestion probability in a single delay system with a Pareto input stream and constant service time*

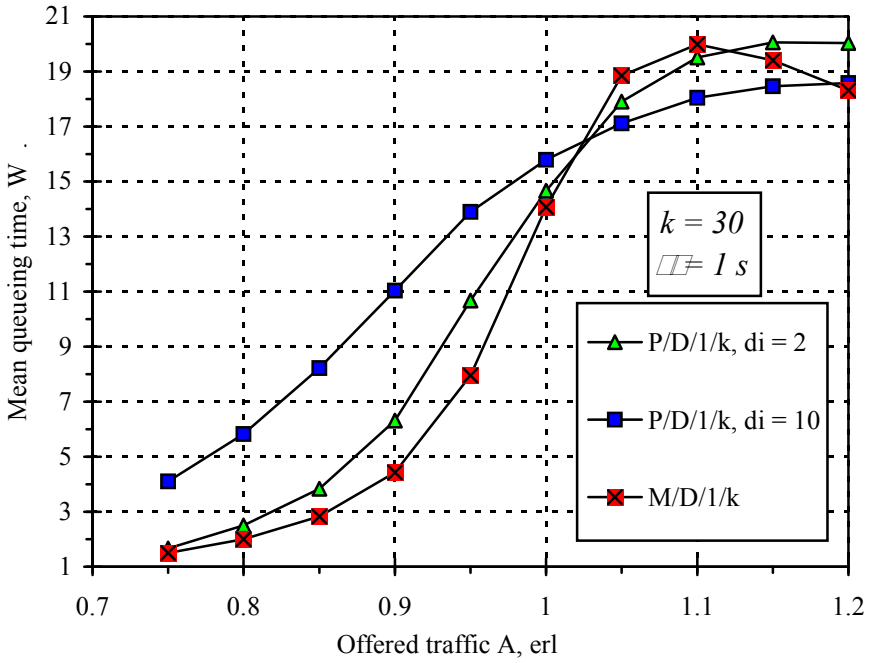


Figure 4. *Mean system time in seconds in a single delay system with a Pareto input stream and constant service time*

In this paper, a generalised Pareto distribution is introduced and explained. A basic simulation model for a queueing system Pareto/D/1/k is examined in detail. The developed simulation model provides a unified framework to model peak input traffic. Numerical results and subsequent experience have shown that this model is accurate and useful in analyses of teletraffic systems.

The importance of a single server queue in a case of a Pareto input stream and constant service time comes from its ability to describe behaviour that is to be found in more complex real queueing systems. It is one of the cases in a general teletraffic system that is important in telecommunication systems design.

In conclusion, we believe that the presented simulation model will be useful in practice.

Our model permits us to look at the queueing behaviour. We saw that as the load increases, the long-tailed nature of the queue brings to big losses and delay. Comparisons with Poisson arrivals showed that the simple Markovian models seriously underestimate the performance of such systems. In a sense, our results help solidify those statements being made by other authors.

The simulation method we have presented could certainly be used to study congestion in the Next Generation Networks. Our method generates a complete probabilistic analysis of the queues we study. The method is quick and its accuracy can be easy evaluated. We have used the method with the Pareto only, but are investigating its use with other distributions.

We feel that our simulation method has excellent promise to analyze the type of congestion problems and delays seen on the Internet. Thus, we are continuing our research using the simulation method for a larger class of queueing systems.

Acknowledgements

This paper is sponsored by the National Science Funds of MES - Bulgaria in the framework of project **BY-TH-105/2005** “Multimedia Telecommunications Networks Planning with Quality of Service and Traffic Management”.

Bibliography

[Argibay, 2003] Argibay Losada P., A. Suárez González, C. López García, R. Rodríguez Rubio, J. López Ardao, D. Teijeiro Ruiz. On the simulation of queues with Pareto Service. Proc. 17th European Simulation Multiconference, pp. 442-447, 2003.

[Cao, 2004] Cao J., W. Cleveland, and D. Sun. Bandwidth estimation for best-effort Internet traffic. Statist. Sci., Volume 19, Number 3 (2004), pp. 518-543.

[Chlebus, 2007] Chlebus E., Gautam Divgi. The Pareto or Truncated Pareto Distribution? Measurement-Based Modelling of Session Traffic for Wi-Fi Wireless Internet Access. Wireless Communications and Networking Conference - WCNC, IEEE, pp. 3625-3630, 2007.

[Eickhoff, 2006] Eickhoff M., D. McNickle, and K. Pawlikowski. Analysis of the time evolution of quantiles in simulation. International Journal of Simulation, Vol. 7, No 6, pp. 44-55, 2006.

[Fernandes, 2003] Fernandes S., C. Kamienski, D. Sadok. Accuracy and Computational Efficiency on the Fractal Traffic Generation. Proc. of the 3rd IASTED International Multi-Conference on Wireless and Optical Communications - WOC, 2003.

[Fischer, 2005] Fischer M.; D. Bevilacqua Masi, D. Gross, J. Shorte. One-Parameter Pareto, Two-Parameter Pareto, Three-Parameter Pareto: Is There a Modelling Difference?. Telecommunications Review, pp. 79-92, 2005.

[Fischer, 1999] Fischer M.; H. Cart. A Method for Analyzing Congestion in Pareto and Related Queues. Telecommunications Review, pp. 15-27, 1999.

[Gross, 2003] Gross D., M. Fischer, D. Masi, J. Shorte. D. Gross. Difficulties in Simulating Queues with Pareto Service. Proceedings of the Winter Simulation Conference, pp. 407-415, 2003.

[Inmaculada, 2006] Inmaculada A., M. Meerschaert, and A. Panorska. Parameter Estimation for the Truncated Pareto Distribution, Journal of the American Statistical Association, Volume 101, Number 473, pp. 270-277, 2006.

[Koh, 2003] Koh Y.; Kiseon Kim. Loss probability behaviour of Pareto/M/1/K queue. Communications Letters, IEEE, Volume 7, Issue 1, pp. 39-41, Jan 2003.

[Koh, 2003] Koh Y.; Kiseon Kim. Evaluation of Steady-State Probability of Pareto/M/1/K Experiencing Tail-Raising Effect. Lecture Notes in Computer Science, Volume 2720, pp. 561-570, 2003.

[Rodrıguez, 2004] Rodrıguez-Dagnino R.. On the Pareto/M/c and Pareto/M/1/K. queues. Proc. SPIE, ITCom, vol. 5598, pp. 183-193, 2004.

[Salvador, 2004] Salvador P., A. Pacheco and R. Valadas. Modelling IP traffic: joint characterization of packet arrivals and packet sizes using BMAPs. Computer Networks, Volume 44, Issue 3 , 2004, pp. 335-352.

[Khadjiivanov, 1993] Khadjiivanov L., B.T.Taskov, A.A.Aliazidi, B.P.Tsankov. Application of Priority Queueing Mechanisms to ATM Multiplexing and Traffic Control. Proc. of Integrated Broadband Communications Networks and Services, Copenhagen, Denmark, April 20 – 23, 1993, pp. 33.3.1 – 33.3.11.

[Tsankov, 2007] Tsankov B., R. Pachamanov, and D. Pachamanova. Modified Brady Voice Traffic Model for WLAN and WMAN. Electronics Letters, vol.43, issue 23, Nov. 2007, pp. 1295-1297.

[Zhang, 2007] Zhang W.; He Jingsha. Modeling End-to-End Delay Using Pareto Distribution. Second International Conference on Internet Monitoring and Protection, ICIMP, pp. 21-21, 2007.

Authors' Information

Seferin Mirtchev – *Technical University of Sofia, Kliment Ohridski St., N:8, Bl.1, Sofia-1000, Bulgaria;*
e-mail: stm@tu-sofia.bg

Rossitza Goleva – *Technical University of Sofia, Kliment Ohridski St., N:8, Bl.1, Sofia-1000, Bulgaria;*
e-mail: rig@tu-sofia.bg

PRIMARY AND SECONDARY EMPIRICAL VALUES IN NETWORK REDIMENSIONING

Emiliya Saranova

***Abstract**: A model of an overall telecommunication network with virtual circuits switching, in stationary state, with Bernoulli-Poisson-Pascal (BPP) input flow, repeated calls, limited number of homogeneous terminals and 8 types of losses is considered. One of the main problems of network redimensioning is estimation of the traffic offered in the network because it reflects on finding of necessary number of equivalent switching lines on the basis of the consideration of detailed users behavior and target Quality of Service (QoS).*

The aim of this paper is to find a new solution of Network Redimensioning Task (NRDT) [4], taking into account the inconvenience of necessary measurements, not considered in the previous research [5].

The results are applicable for redimensioning of every (virtual) circuit switching telecommunication system, both for wireline and wireless systems (GSM, PSTN, ISDN and BISDN). For packet - switching networks proposed approach may be used as a comparison basis and when they work in circuit switching mode (e.g. VoIP).

***Keywords**: Overall Network Traffic, Offered Traffic, Virtual Circuits Switching.*

***ACM Classification Keywords**: C.2.1 Network Architecture and Design; C.2.3 Network Operations; C.4 Performance of Systems.*

Introduction

The task of Teletraffic engineering, often considered in real telecommunication system, is to find dependencies between three basic quantities: traffic demand, quality of services (QoS) and technical parameters of the servicing system [6]. An estimation of some teletraffic parameters (offered traffic intensity, incoming rate of the first call attempts, etc.) in the network is one of the main problems of network redimensioning [2]. Network redimensioning [3] is necessary for medium term traffic management in an advance determined level. Based on the ITU definitions 4.1, 4.2, 2.8 and 2.11 in [1], of QoS parameters, we use the following two parameters, dependable from the network macro-state (*Yab – traffic of all network terminals*): probability *Pbs* (blocked switching) due to lack of resources, and probability *Pbr* of finding B-terminals busy. We denote the target value of blocked switching by *trg.Pbs*.

In this paper we consider detailed conceptual and its corresponded analytical traffic model [4] of telecommunication system with channel switching, in stationary state, with generalized BPP input flow, repeated calls, limited number of homogeneous terminals and losses due to abandoned and interrupted dialing, blocked and interrupted switching, not available intent terminal, blocked and abandoned ringing and abandoned conversation.

A system of equations based on the conceptual model and some dependencies between parameters of the researched telecommunication system, is derived. An analytical solution of a network redimensioning task (NRDT) and the necessary conditions for it are researched.

The results are useful for finding of suitable method for estimation of the necessary number of equivalent internal switching lines (*Ns*) in dimensioning and redimensioning tasks.

Conceptual model and analytical models

The conceptual model (shown on Fig. 1) [4] of the telecommunication system includes the paths of the calls, generated from (and occupying) the A-terminals in the proposed network traffic model and its environment.

The names of the devices are constructed according to their position in the model.

2.1. The comprising virtual devices

The following important comprising virtual devices are shown on Fig.1:

a = comprises all the A-terminals (calling) in the system (shown with continuous line box).

b = comprises all the B-terminals (called) in the system (box with dashed line).

ab = comprises all the terminals (calling and called) in the system (not shown on Fig.1);

s = virtual device corresponding to the switching system. It is shown with dashed line box into the a - device. *Ns* stand for the capacity (number of equivalent internal switching lines) of the switching system.

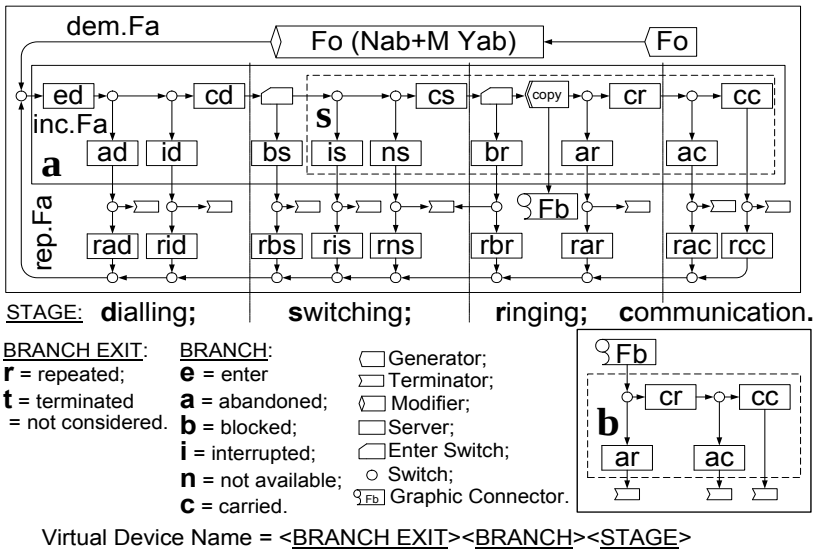


Fig. 1. Normalized conceptual model of the telecommunication system and its environment and the paths of the calls, occupying A-terminals (a - device), switching system (s - device) and B-terminals (b - device); base virtual device types, with their names and graphic notation.

2.2. Stages and branches in the conceptual model:

Considered service *stages*: **d**ialing, **s**witching, **r**inging and **c**ommunication.

Every service stage has *branches*: **e**nter, **a**bandoned, **b**locked, **i**nterrupted, **n**ot available, **c**arried (correspondingly to the modeled possible cases of ends of the calls' service in the branch considered).

Every branch has two *exits*: **r**epeated, **t**erminated (which show what happens with the calls after they leave the telecommunication system). Users may make a new bid (repeated call), or to stop attempts (terminated call).

2.3. Parameters and its notations in the conceptual model:

F = the calling rate (frequency) of the bids' flow [calls/sec.], *P* = probability for directing the calls of the external flow to the device considered, *T* = mean service time, in the device [sec.], *Y* = intensity of the device traffic [Erl], *N* = number of service places (lines, servers) in the virtual device (capacity of the device). In the normalized models [4], used in this paper, every base virtual device, except the switch, has no more than one entrance and/or one exit. Switches have one entrance and two exits. For characterizing the calling rate of the flow, we are using the following notation: *inc.F* for incoming flow, *dem.F*, *ofr.F* and *rep.F* for demand, offered and repeated flows respectively [4]. The same characterization is used for traffic intensity (*Y*).

Fo is the demand calling rate of first call attempts of one idle terminal; *inc.Fa* = *Fa* is calling rate of incoming flow; dem.Fa *is the* calling rate of all demand calls, *M* is modifier of incoming flow.

For creating a simple analytical model, a system of fourteen assumptions is made [4].

Analytical model

Some general equations

For the proposed conceptual model we have derived the following system of equations [4]:

$$Yab = Fa[S_1 - S_2(1 - Pbs)\,Pbr - S_3\,Pbs] \quad (3.1)$$

$$Fa = dem.Fa + rep.Fa \quad (3.2)$$

$$dem.Fa = Fo\,(Nab + M\,Yab) \quad (3.3)$$

$$rep.Fa = Fa\,[R_1 + R_2\ Pbr\,(1-Pbs) + R_3\ Pbs] \tag{3.4}$$

$$Pbr\ = \begin{cases} \dfrac{Yab\text{-}1}{Nab\text{-}1} & in\ \ case\ of\ \ \ 1 \le Yab \le Nab, \\ 0 & in\ case\ of\ \ \ 0\ \le Yab < 1. \end{cases} \tag{3.5}$$

$$Ts = S_{1Z} - S_{2Z}\ Pbr \tag{3.6}$$

$$ofr.Fs = Fa\,(1\text{-}\,Pad)(1\text{-}\,Pid) \tag{3.7}$$

$$ofr.Ys = ofr.Fs\ \ Ts \tag{3.8}$$

$$Erl\_b(Ns, ofr.Ys) = \frac{\dfrac{(ofr.Ys)^{Ns}}{Ns!}}{\sum_{j=0}^{Ns} \dfrac{(ofr.Ys)^{j}}{j!}} \tag{3.9}$$

$$crr.Ys = (1-Pbs)\ \ ofr.Ys \tag{3.10}$$

The following notations are used:

$$S_1 = Ted + Pad\ Tad + (1-Pad)[Pid\ Tid + (1-Pid)[Tcd + Pis\ Tis + (1-Pis)[Pns\ Tns + (1-Pns)[Tcs + 2\,Tb]]]] \tag{3.11}$$

$$S_2 = (1-Pad)(1-Pid)(1-Pis)(1-Pns)[2\,Tb - Tbr] \tag{3.12}$$

$$S_3 = (1-Pad)(1-Pid)[PisTis - Tbs + (1-Pis)[Pns\,Tns + (1-Pns)[Tcs + 2\,Tb]]] \tag{3.13}$$

$$S_{1z} = Pis\,Tis + (1-Pis)[PnsTns + (1-Pns)(Tb + Tcs)] \tag{3.14}$$

$$S_{2z} = (1-Pis)(1-Pns)(Tb - Tbr) \tag{3.15}$$

$$R_1 = Pad\ \mathrm{Pr}\,ad + (1-Pad)(Pid\ \mathrm{Pr}\,id + (1-Pid)[Pis\ \mathrm{Pr}\,is + (1-Pis)(Pns\ \mathrm{Pr}\,ns + (1-Pns))Q]) \tag{3.16}$$

$$R_2 = (1-Pad)(1-Pid)(1-Pis)(1-Pns)(\mathrm{Pr}\,br - Q) \tag{3.17}$$

$$R_3 = (1-Pad)(1-Pid)\{\mathrm{Pr}\,bs - [Pis\ \mathrm{Pr}\,is + (1-Pis)[Pns\ \mathrm{Pr}\,ns + (1-Pns)Q]]\} \tag{3.18}$$

$$Q = Par\,Prar + (1-Par)[Pac\,Prac + (1-Pac)\,Prcc] \tag{3.19}$$

Right Teletraffic Tasks Parameters

Full parameters' set

In the conceptual model presented, we have 31 base and 4 (*a*, *b*, *ab* and *s*) comprising virtual devices. Since every device has 5 parameters (P, F, T, Y, N), the total sum of parameters in our model is 175.

Base parameters' set

There are many obvious dependencies in a system tuple [4], corresponding to the Full Parameters' Set of the Conceptual Model. For example, the sum of probabilities of outgoing transitions in every virtual switch devices has value one; in stationary state Little's formula (Y = F T) is in force for every virtual device; we assume most of devices with infinite capacity. As a result, there are sets of base parameters (sub-tuples), with the following property: If we know the values of the base parameters, we may calculate the values of all other parameters of the same system tuple. Several different base parameters' sets may exist. After careful analysis [4] we have chosen a base parameters' set with 41 parameters.

Parameters' classification based on characterized entities

The 41 parameters of the chosen base parameters' set may be classified, according characterized entities, in the five following groups, corresponding to:

1. Human Behaviour Parameters are 21: *Fo, Nab, Prad, Tid, Prid, Pris, Tis, Pns , Tns , Prns, Tbs, Prbs, Tbr, Prbr, Par, Tar, Prar, Tcr, Prac, Tcc, Prcc*;
2. Technical Characteristics Parameters are 4: *Pid, Pis, Tcs*, *Ns*;

3. Mix Factors' Parameters are 6: *Ted, Pad, Tad, Tcd, Pac, Tac*;
4. Modeller Chosen Values Parameter (1): *M*;
5. Derived Parameters from the previous four groups are 9: *Yab*, *Fa, dem.Fa, rep.Fa, Pbs, Pbr, ofr.Fs*, *Ts, ofr.Ys.*

Parameters' classification based on their values' determination and its notations

We consider

1. Administratively determined values with
   - **Target parameters' values:** denoted by prefix *trg.*, e.g. *trg.Pbs* is a target values of blocking probability due to lack of sufficiency switching lines*;*

2. Parameters empirical evaluated are called

   Primary:

   - If empirical parameters' values are received after direct measurements, denoted by prefix *emp.*, e.g. *emp.crr.Ys* is empirical values of carried traffic intensity.

   Secondary:

   - Parameters with empirical values, received from primary after calculation: For example, *emp.Fo*, *emp.Yab, emp.crr.Ys;*
   - Designed parameters and their values**:** denoted by *dsn.,* e.g. *dsn.ofr.Ys, dsn Ts;*
   - Thresholds values of parameters are received as restriction in the dependencies, denoted by *thr*, i.e. *thr.Fo.*

Static and Dynamic Parameters' Classification

In this paper, we propose a short term classification of the chosen base parameters' set with 31 static and 10 dynamic parameters.

For the static parameters we assume that their values don't depend on the state of the system and correspondingly on the intensity of the input flow. They may depend on other factors, e.g. the time of the day; seasons, human temperament, Telecom Administration, Gross Domestic Product and so on, but for the observed and modelled time interval we consider them as constants.

The 10 dynamic parameters, with mutually dependent values are: *Fo Yab, Fa, dem.Fa, rep.Fa, Pbs, Pbr, ofd.Fs, Ts, ofd.Ys*.

Finding Terminal Teletraffic Parameters

Task Formulation:

We consider the overall telecommunication system conceptual model, presented in Fig. 1 and described in Section 2. Parameters with known values are all the P (probability for call direction) and T (holding time) parameters of the base virtual devices, plus values of the intensity of incoming calls flow (*Fa*) and traffic carried (*crr.Ys*).

Known parameters' values:

Parameters with empirical values:

Primary: $emp.crr.Ys = crr.Ys,\ emp.inc.Fa = Fa,\ emp.Tb = Tb$ (5.1)

Secondary: $S_1, S_2, S_3, R_1, R_2, R_3, S_{1z}, S_{2z}, Pad, Pid, Pbs$ (5.2)

Aim: Based on previous experience, to determine the incoming input flow, generated from one idle terminal and probability of finding B – terminal busy in the real system;

Unknown parameters' values: $emp.Fo,\ emp.Pbr$ (5.3)

We assume the values of parameters $emp.Fo, S_1, S_2, S_3, R_1, R_2, R_3, S_{1z}, S_{2z}, Pad, Pid, Pbs$ as mutually independent and having the same values before and after changing the number of switching lines.

5.2. Analytical Solution:

Theorem 1: Empirical values of $emp.Pbr$ fulfills the dependence:

$$emp.Pbr = \frac{S_{1z}}{S_{2z}} - \frac{crr.Ys}{S_{2z}\, Fa\,(1-Pad)\,(1-Pid)\,(1-Pbs)} \tag{5.4}$$

Proof: From equations (3.7), (3.8), (3.10), if $ofr.Ys < S_{1z}\, Fa\,(1-Pad)\,(1-Pid)$ follows

$$crr.Ys = Fa\,(1-Pad)\,(1-Pid)\,(1-Pbs)\,Ts \tag{5.5}$$

We receive from equations (3.9) (3.6) and (5.5), regarding $emp.Pbr$,the expression (5.4).

Theorem 2: The follow expression regarding $emp.Fo$ exist

$$emp.Fo = \frac{Fa\,(1-Pbs)\left\{(1-R_1-R_3\,Pbs)\,S_{2z} - R_2\,\Omega\right\}}{S_{2z}\,(Nab+M)\,(1-Pbs) + M\,(Nab-1)\,\Omega}, \text{ where}$$

$$\Omega = S_{1z}\,(1-Pbs) - \frac{crr.Ys}{Fa\,(1-Pad)(1-Pid)}\ . \tag{5.6}$$

Proof: If $S_{2z} \neq 0$ then from equations (3.2) and (3.4) follows

$$dem.Fa = Fa\left\{1 - R_1 - R_3 Pbs - R_2(1-Pbs)\,emp.Pbr\right\} \tag{5.7}$$

We receive from equations (3.3) and (3.5)

$$dem.Fa = emp.Fo\left\{Nab + M + M\,(Nab-1)\,emp.Pbr\right\} \tag{5.8}$$

Based on (5.7), (5.8) and the proved dependence in Theorem 1 (5.4), we determine (5.6).

Conclusion: The evaluation of, calls flow generated from one idle terminal ($emp.Fo$), is based on carried traffic intensity $crr.Ys$ and calling rate of the input flow Fa . These parameters are easily measurable. Others parameters as $S_1, S_2, S_3, R_1, R_2, R_3$ are measurable with difficulty, in principle, but they are considered as independent of the input flow and steady approximately, i.e. as constants.

Therefore, based on previous experience and the above specifications, parameters' evaluation with one measurement of some primary parameter values in a short time interval is possible. It is make a network redimensioning task solution easier.

Network Redimensioning Task

Based on previous experience, determining the volume of telecommunication resources that is sufficient to serve a given input flow, with prescribed characteristics of QoS, is one of the main problems that often have to be solved by network operators. It includes the following sub-tasks:

1. Redimensioning a network means to be found of number of equivalent internal switching lines necessary to satisfy a level of QoS that has been administratively pre-determined.
2. Finding the values of the designed parameters, describing the designed system state, based on known and target parameters' values. For example, a system parameter, describing offered traffic intensity of the switching system (*dsn.ofr.Ys*), designed probability to find B terminal "busy" (*dsn.Pbr*), etc…

Analytical Solution of a NRDT on Basis of Easy Measurable Empirical Values

On the basis of equation (5.6) from this paper and equations (7.2.4) and (7.2.10) from [5], the Network redimensioning task solution, using easy measurable parameters, we obtain the following two equations:

$$dsn.ofr.Ys = \begin{cases} \dfrac{emp.Fo\ Nab\ (1-Pad)(1-Pid)(S_{1z}-S_{2z}\ dsn.Pbr)}{1-R_1-R_3\ trg.Pbs-emp.Fo\ M(S_1-S_3\ trg.Pbs)}, & if \quad 0 \le Fo \le thr.Fo \\ \dfrac{emp.Fo\ \ Nab(1-Pad)(1-Pid)(S_{1z}-S_{2z}\ dsn.Pbr)}{1-R_1-emp.FoMS_1+(emp.FoMS_2-R_2)(1-trg.Pbs)\ dsn.Pbr+(emp.FoMS_3-R_3)trg.Pbs}, & \\ & if \quad Fo \ge thr.Fo. \end{cases}$$

and the equation $trgPbs = Erl\_b(Ns, dsn.ofr.Ys)$. (5.9)

The expression *Erl_b (Ns, dsn.ofr.Ys)* is the famous Erlang B - formulae.

It is proved in Theorem 7.4 in [5] that only one solution of *Ns* exists, fulfilling the equation (5.9) and corresponding to the determined administratively in advance value of the blocking probability *trg.Pbs* $\in (0; 1]$.

Numerical Results

Computer program on the basis of empirical values, received from the new parameters dependencies (5.6), (5.8) and (5.9) is worked out.

Verification of the new parameters dependencies is made and the maximal absolute difference between the real and calculated values is less than 1.561×10^{-17} in the whole admissible interval.

Conclusions

New dependencies between empirical values of carried traffic intensity $emp.crr.Ys$ and calling rate $emp.Fa$, from one hand, and the demand rate of calls of one idle terminal $emp.Fo$, from other hand, are derived.

Based on the new dependencies, network redimensioning task is solved using easy measurable values of empirical parameters.

The new parameters dependencies are numerical verified **in** the whole admissible interval.

The received results make the network dimensioning/redimensioning, based on QoS requirements easier. The described approach is applicable for every (virtual) circuit switching telecommunication system (like GSM and PSTN) and may help considerably for ISDN, BISDN and most of core and access networks dimensioning. For packet switching systems, proposed approach may be used when they work in circuit switching mode.

Bibliography

1. ITU-T Recommendation E.600: Terms and Definitions of Traffic Engineering. Melbourne, 1988; revised at Helsinki, 1993
2. ITU-T Recommendation E.501: Estimation of traffic offered in the network. (revised 26. May 1997);
3. ITU-T Recommendation E.734 Methods for allocating and dimensioning Intelligent Network (IN) resources, October 1996
4. S. A. Poryazov, E. T. Saranova. , 2006. Some General Terminal and Network Teletraffic Equations in Virtual Circuit Switching Systems. Chapter in: A. Nejat Ince, Ercan Topuz (Editors). "Modelling and Simulation Tools for Emerging Telecommunications Networks: Needs, Trends, Challenges, Solutions", Springer Sciences+Business Media, LLC 2006, pp. 471-505. Printed in USA, Library of Congress Control Number: 2006924687. ISBN-13: 978-0387-32921-5 (HB)
5. Saranova E. T., 2006. Redimensioning of Telecommunication Network based on ITU definition of Quality of Services Concept, In: Proceedings of the International Workshop "Distributed Computer and Communication Networks", Sofia, Bulgaria, 2006, Editors: V. Vishnevski and Hr. Daskalova, Technosphera publisher, Moscow, Russia, 2006, ISBN 5-85638-111-4 pp. 165 - 179;
6. Цанков Б. ,2006. Телекомуникации фиксирани, мобилни и IP, изд. Нови знания, ТУ - София, 2006.ISBN -10: 954-9315-58-4, ISBN – 13: 978-954-9315-58-5

Author's Information

Emiliya Saranova – e-mail: saranova@hctp.acad.bg, Emiliya@cc.bas.bg
Institute of Mathematics and Informatics - Bulgarian Academy of Science, Sofia, Bulgaria
College of Telecommunication and Posts, Sofia, Bulgaria

IMPLEMENTATION OF A HEURISTIC METHOD OF DECOMPOSITION OF PARTIAL BOOLEAN FUNCTIONS

Arkadij Zakrevskij, Nikolai Toropov

***Abstract**: An original heuristic algorithm of sequential two-block decomposition of partial Boolean functions is researched. The key combinatorial task is considered: finding of suitable partition on the set of arguments, i. e. such one, on which the function is separable. The search for suitable partition is essentially accelerated by preliminary detection of its traces. Within the framework of the experimental system the efficiency of the algorithm is evaluated, the boundaries of its practical application are determined.*

***Keywords**: Partial Boolean Functions, Heuristic Method of Decomposition*

***ACM Classification Keywords**: B.6. Combinational logic, switching theory, automatic synthesis, optimization.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

It is generally accepted to understand decomposition of a Boolean function as presenting it by a composition of functions of smaller number of variables. This task is diversiform - enough to note, that any nontrivial logic circuit, implementing some Boolean function of many variables, can be considered as a composition of functions realizable by separate units. We shall consider a private but important case of the task becoming classical, as not one hundred publications is devoted to it [1], namely *sequential two-block decomposition*.

In this case the problem is stated as follows. On the set of arguments $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ a Boolean function $f(\boldsymbol{x})$ is preset. It is necessary to replace $f(\boldsymbol{x})$ by an equivalent composition $g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v})$ of Boolean functions $g(h, \boldsymbol{w}, \boldsymbol{v})$ and $h(\boldsymbol{u}, \boldsymbol{w})$ of smaller number of variables. By that the conditions $\boldsymbol{x} = \boldsymbol{u} \cup \boldsymbol{w} \cup \boldsymbol{v}$ and $\boldsymbol{u} \cap \boldsymbol{w} = \boldsymbol{u} \cap \boldsymbol{v} = \boldsymbol{w} \cap \boldsymbol{v} = \varnothing$ should be fulfilled, generating a *weak partition* $\boldsymbol{u}/\boldsymbol{v}$ on the set of arguments $\boldsymbol{x}$. This replacement is named as decomposition of the function $f(\boldsymbol{x})$ at the partition $\boldsymbol{u}/\boldsymbol{v}$ ($f(\boldsymbol{x})$ is *decomposable* at $\boldsymbol{u}/\boldsymbol{v}$) and can simplify a logic circuit implementing function $f(\boldsymbol{x})$ (for example, at logical synthesis in the basis of units LUT (look up tables)). It is meaningful under the condition ($|\boldsymbol{u}| > 1$) & ($|\boldsymbol{v}| > 0$), otherwise decomposition is trivial – the circuit does not become simpler. If $|\boldsymbol{w}| = \varnothing$, the composition is named *disjunct*, otherwise – *non-disjunct*. The task becomes essentially complicated, if the Boolean function $f(\boldsymbol{x})$ appears to be *partial*, being defined not on all elements of the Boolean space $M = \{0, 1\}^n$. The decomposition of a Boolean function is a difficult combinatorial task, which complexity fast grows with increase of the number of variables n. An original heuristic decomposition technique is considered below, which efficiency is provided practically with acceptable compromise between speed of finding of a solution and its reliability. The program implementation of that method is based on usage of the set of special macro operations above long (2^n-component) Boolean vectors $\boldsymbol{f}$, with which it is possible to represent arbitrary Boolean functions $f(\boldsymbol{x})$ of n variables.

In paper [2] was proved

The assertion 1. At equiprobable sampling of the function $f(\boldsymbol{x})$ from the set of all Boolean functions of n variables the probability of decomposability of this function is bounded above by the value

$$C_n^2 (n-2) \gamma 2^{n-3}, \text{ where } \gamma = 88/256.$$

This value fast tends to zero with growth of n (for example, at n = 6, 12, 18 it receives accordingly values 0,012, 10^{-234}, 10^{-15193}), whence follows, that decomposability of arbitrary Boolean functions of many variables is rather improbable. Therefore, a practical sense the task of decomposition has only for such function, which, as a priory

is known, is presented by some composition, which is required to be found. The task in this setting is considered below.

Preparation of input data

An approach is widely spread, according to which concrete examples of input data for programs of solution of the diversiform tasks of logical design are selected from special libraries shaped in view of some practical reasons. In the given paper another way is offered based on generation of random examples with set values of some parameters.

In the considered task of decomposition such parameters are: n – the number of arguments of the function $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_n)$, $p = |\boldsymbol{u}|$ - (the power of set $\boldsymbol{u}$), $q = |\boldsymbol{v}|$ and r - the degree of uncertainty of the function $f(\boldsymbol{x})$ (more exactly this value will be defined below).

During preparation of an example there are selected by random from the set $\boldsymbol{x}$ p variables forming the set $\boldsymbol{u}$, and q variables forming the set $\boldsymbol{v}$. Remaining variables of the set $\boldsymbol{x}$ will form the set $\boldsymbol{w}$. Then random Boolean functions $h(\boldsymbol{u}, \boldsymbol{w})$ and $g(x, \boldsymbol{w}, \boldsymbol{v})$ are generated. These operations are simple enough and are fulfilled with usage of generators of random evenly distributed Boolean vectors.

More composite appears the composition of functions $h(\boldsymbol{u}, \boldsymbol{w})$ and $g(x, \boldsymbol{w}, \boldsymbol{v})$, resulting in obtaining the function $f(\boldsymbol{x})$. At presentation of the offered below method of solving this task it is convenient to use the language of macro operations above long 2^n-component Boolean vectors $\boldsymbol{f}$. Along with standard component-wise operations above vectors of identical dimension ($\boldsymbol{a} \vee \boldsymbol{b}$, $\boldsymbol{a} \oplus \boldsymbol{b}$, etc) it contains high-performance operations above adjacent elements of Boolean space, fulfilled in parallel in all 2^{n-1} couples of adjacent elements [3]. While we shall take advantage from following of such operations:

$\boldsymbol{f} - k$ – assignment of value 0 to argument x_k (obtaining of the function $f(x_k = 0)$),

and also generalizing it operation $\boldsymbol{f} - \boldsymbol{u}$, at which execution the value 0 is assigned to all arguments, marked with 1 in n-component vector $\boldsymbol{u}$.

We shall use also the operation $\boldsymbol{h} \times \boldsymbol{u}$ of mapping the function $h(\boldsymbol{u})$ onto the interval of space M, corresponding to the conjunction of inversions of the variables not marked in $\boldsymbol{u}$. All elements of remaining intervals with the same outer variables gain by that the value 0.

In terms of these operations the program of calculation of vector $\boldsymbol{f}$, representing the required function $f(\boldsymbol{x})$, looks so:

$$\boldsymbol{a} := \boldsymbol{h} \times (\boldsymbol{u}, \boldsymbol{w}) - \boldsymbol{v},$$
$$\boldsymbol{b} := \boldsymbol{g}_0 \times (\boldsymbol{w}, \boldsymbol{v}) - \boldsymbol{u},$$
$$\boldsymbol{c} := \boldsymbol{g}_1 \times (\boldsymbol{w}, \boldsymbol{v}) - \boldsymbol{u},$$
$$\boldsymbol{f} := \bar{\boldsymbol{a}}\, \boldsymbol{b} \vee \boldsymbol{a}\, \boldsymbol{c},$$

where the vectors $\boldsymbol{g}_0$ and $\boldsymbol{g}_1$ are represented with two halves of vector $\boldsymbol{g}$, specifying coefficients of decomposition of the function $g(x, \boldsymbol{w}, \boldsymbol{v})$ by variable x.

Let's illustrate this program on an example (fig. 1), where $n = 6$, and the sets $\boldsymbol{u} = (x_1, x_2)$, $\boldsymbol{w} = (x_3, x_4)$ and $\boldsymbol{v} = (x_5, x_6)$ are preset by appropriate Boolean vectors $\boldsymbol{u} = 110000$, $\boldsymbol{w} = 001100$ and $\boldsymbol{v} = 000011$.

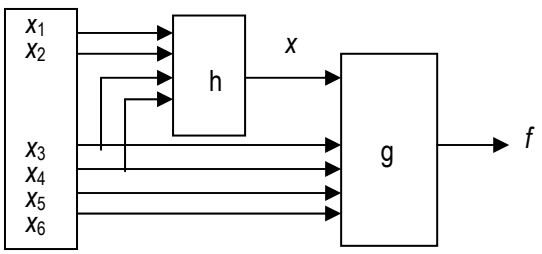


Fig.1

We admit, that the generated Boolean functions $h(\boldsymbol{u}, \boldsymbol{w})$ and $g(x, \boldsymbol{w}, \boldsymbol{v})$ are represented accordingly by 2^4-component vector $\boldsymbol{h}$ and 2^5-component vector $\boldsymbol{g}$ with the generally accepted order of following of components:

$\boldsymbol{h}$ = 11010010 01101100,

$\boldsymbol{g}$ = 00110100 11001001 10100101 10101011.

Then the calculations are reduced to obtaining the shown below sequence of Boolean vectors. At calculation of vectors $\boldsymbol{a}^* = \boldsymbol{h} \times (\boldsymbol{u}, \boldsymbol{w})$, $\boldsymbol{b}^* = \boldsymbol{g}_0 \times (\boldsymbol{w}, \boldsymbol{v})$ and $\boldsymbol{c}^* = \boldsymbol{g}_1 \times (\boldsymbol{w}, \boldsymbol{v})$ the appropriate interval of space M is found at first (it is marked by the bold font). Then the vector of the considered function ($\boldsymbol{h}$, $\boldsymbol{g}_0$ or $\boldsymbol{g}_1$) is brought into it, with saving the order of components following.

```
                                 -------- -------- -------- -------- x1
                 -------- --------                  -------- --------   x2
          -------          --------          --------          -------- x3
     ----      ----      ----      ----      ----      ----      ----      ----   x4
   --  --    --  --    --  --    --  --    --  --    --  --    --  --    --  -- x5
 - - - -   - - - -   - - - -   - - - -   - - - -   - - - -   - - - -   - - - -   x6

10001000 00001000 00000000 10000000 00001000 10000000 10001000 00000000  a*
11111111 00001111 00000000 11110000 00001111 11110000 11111111 00000000  a
00110100 11001001 00000000 00000000 00000000 00000000 00000000 00000000  b*
00110100 11001001 00110100 11001001 00110100 11001001 00110100 11001001  b
10100101 10101011 00000000 00000000 00000000 00000000 00000000 00000000  c*
10100101 10101011 10100101 10101011 10100101 10101011 10100101 10101011  c
00000000 11000000 00110100 00001001 00110000 00001001 00000000 11001001  ab
10100101 00001011 00000000 10100000 00000101 10100000 10100101 00000000  ac
10100101 11001011 00110100 10101001 00110101 10101001 10100101 11001001  f
```

After obtaining the function $f(\boldsymbol{x})$ an uncertainty is brought into it by replacement of some of its values (0 or 1) with the symbol of uncertainty "–". The result of this conversion is represented by one ternary vector $\boldsymbol{f}^-$ or the couple of Boolean vectors $\boldsymbol{f}^0$ and $\boldsymbol{f}^1$, in which unities mark values 0 and 1 of function $f(\boldsymbol{x})$. For example (n = 4),

$\boldsymbol{f}^-$ = 011–0001 –101–110,

$\boldsymbol{f}^0$ = 10001110 00100001,

$\boldsymbol{f}^1$ = 01100001 01010110.

The degree of uncertainty of the function is set by the parameter r, receiving values from set {0, 1, 2, …, 31} and defining probability $r/32$ that the arbitrary selected component of vector $\boldsymbol{f}^-$ will receive value "–".

Depositing of uncertainty in the function $f(\boldsymbol{x})$ is carried out on the basis of results obtained in [4], which essence can be caught from the following example. In order to receive a random Boolean vector with probability 19/32 of appearance of unity in an arbitrary selected component, it is enough to present number 19 by its binary code 10011, then to put in correspondence to values 0 and 1 Boolean operators $\wedge$ and $\vee$, and, sorting out components of the code from the right to the left, to fulfill the following sequence of operations above completely random (with probability 0,5 of appearances of 1 in any component) independent Boolean vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_4, \boldsymbol{c}_5$:

$$((\mathbf{0} \vee \boldsymbol{c}_1 \vee \boldsymbol{c}_2) \wedge \boldsymbol{c}_3 \wedge \boldsymbol{c}_4) \vee \boldsymbol{c}_5$$

where **0** is a vector, which all components are equal to zero.

Method of search for a suitable partition by traces

In the basis of Boolean functions decomposition techniques the solution of the following tasks lays.

The task 1. For a given function $f(\boldsymbol{x})$ and partition $\boldsymbol{u}/\boldsymbol{v}$ to clarify, whether $f(\boldsymbol{x})$ is decomposable at $\boldsymbol{u}/\boldsymbol{v}$, i. e. whether there exists a composition $g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v})$, equivalent to function $f(\boldsymbol{x})$, and, maybe, to find functions g and h.

If such a composition exists, we shall term partition $\boldsymbol{u}/\boldsymbol{v}$ *suitable*.

The task 2. For a given function $f(\boldsymbol{x})$ to find a suitable partition.

The second task is more difficult. It is obvious, that the exhaustive search of all partitions with the purpose of checking them on fitness practically is unrealizable at major n, as their number is approximated by the value 3^n. Much more efficient is the method of search of suitable partition by traces, offered in [2]. It is based on the following definitions and assertions.

Consider two partitions $\boldsymbol{u}/\boldsymbol{v}$ and $\boldsymbol{u}^*/\boldsymbol{v}^*$, bound with relations $\boldsymbol{u}^* \subseteq \boldsymbol{u}$ and $\boldsymbol{v}^* \subseteq \boldsymbol{v}$. Let's speak, that the partition $\boldsymbol{u}^*/\boldsymbol{v}^*$ *submits* to partition $\boldsymbol{u}/\boldsymbol{v}$, or is its *trace*.

The assertion 2. If the Boolean function $f(\boldsymbol{x})$ is decomposable at partition $\boldsymbol{u}/\boldsymbol{v}$, it is decomposable also at its trace $\boldsymbol{u}^*/\boldsymbol{v}^*$.

Corollary. If the function $f(\boldsymbol{x})$ is not decomposable at partition $\boldsymbol{u}^*/\boldsymbol{v}^*$, it is not decomposable also at $\boldsymbol{u}/\boldsymbol{v}$.

Suppose, that $|\boldsymbol{u}| = k$ and $|\boldsymbol{v}| = m$. A partition with $k = 2$ and $m = 1$ we shall term as a *triad*. It is the simplest of partitions, on which the nontrivial decomposition can take place.

The assertion 3. The Boolean function $f(\boldsymbol{x})$ is not decomposable, if it is not decomposable at any of triads.

Let's start therefore the search for suitable partition $\boldsymbol{u}/\boldsymbol{v}$ from the search for its traces on the set of triads, i.e. from finding of a suitable triad.

Assume, that the search consists in random sampling from a series of q triads completing by finding a suitable one. It is shown in [2], that practically (at $n > 10$) any suitable triad submits to the required partition $\boldsymbol{u}/\boldsymbol{v}$, in other words, accessory solutions miss. At this supposition it is fair

The assertion 4. The expectation $M(q)$ of the value q is equal to $C_n^2(n-2) / C_k^2\, m$.

Thus, the search of a suitable triad is performed rather fast. For example, the formula representing the value $M(q)$ can be approximated by more simple formula $3^3 = 27$ (at $k = m = n/3$) and $2^3 = 8$ (at $k = m = n/2$).

Having found a suitable triad, it is possible to extend sequentially sets $\boldsymbol{u}$ and $\boldsymbol{v}$, testing different variables on possibility of their inclusion into one of these sets (the inclusion is possible, if the extended partition remains suitable).

Programming in macro operations

Fragments of partition $\boldsymbol{u}/\boldsymbol{v}$. Any weak partition $\boldsymbol{u}/\boldsymbol{v}$ divides a 2^n-component ternary vector $\boldsymbol{f}^{-}$ into 2^{n-p-q} parts representing coefficients of disjunctive decomposition of the function $f(\boldsymbol{x})$ by variables of set $\boldsymbol{w}$. Each of these parts can be presented by an appropriate *fragment* – so we shall term a ternary matrix the size $2^p \times 2^q$, which rows correspond to different sets of values of variables from $\boldsymbol{u}$, and columns – to sets of values of variables from $\boldsymbol{v}$.

A partition $\boldsymbol{u}/\boldsymbol{v}$ appears *suitable*, if each its fragment is suitable. And a fragment is *suitable*, if the partial Boolean function $f(\boldsymbol{x})$ can be predetermined in such a way, that the fragment will contain no more than two values of Boolean rows. In other words, the graph of orthogonality of rows of each fragment should be bichromatic [5]. The

check of this condition becomes simpler for a special case of partition – a triad (a, b)/c. In this case the size of the graph equals 4×2.

Checking a triad on fitness. The graph of orthogonality of rows of a triad fragment contains four nodes, therefore, it is bichromatic, if there will be no cycle of length three in it. Consider two rows of a triad fragment corresponding to values (a, b) = (0, 0) and (a, b) = (1, 1).

If such a cycle exists, then at least one of selected nodes will belong to it. Therefore, it is enough to test each of these two nodes on belonging to some cycle of length three. If such belonging will not be revealed, the graph is bichromatic and the triad is suitable. Necessary and sufficient condition of belonging of a node, i.e. the row corresponding to it, to a cycle of length three could be formulated so: among rows orthogonal to the given one, there are mutually orthogonal rows.

The offered way is implemented by the following algorithm, which is remarkable by that it checks on fitness simultaneously all 2^{n-3} fragments corresponding to a given triad, and by that checks fitness of the triad as a whole. The considered partial Boolean function $f(\boldsymbol{x})$ is set by a couple of Boolean vectors $\boldsymbol{f}^0$ and $\boldsymbol{f}^1$.

In this algorithm alongside with introduced earlier operation $\boldsymbol{f} - k$ the operation $\boldsymbol{f} + k$ is used, defined similarly: it means assignment of value 1 to argument x_k. Both operations are easily implemented in Boolean space on couples of elements, adjacent by a variable x_i. The algorithm contains also the operation

$$S_k^* \boldsymbol{f} = (\boldsymbol{f} - k)^* (\boldsymbol{f} + k)$$

of symmetrizations of a vector-function $\boldsymbol{f}$ by variable x_k and Boolean operation $* \in \{\vee, \wedge, \oplus\}$. In further presentation the generalized operation $S_t^* \boldsymbol{f}$ is used, the equivalent to operation $S_k^* \boldsymbol{f}$, which is fulfilled for all variables of a subset $\boldsymbol{t} = (t_1, t_2, \ldots, t_m) \subseteq \boldsymbol{x}$.

Let's illustrate introduced operations by the following examples, in which $\boldsymbol{u} = (x_2, x_5)$:

$$\begin{aligned}
\boldsymbol{f} &= 10010010\ 01111000\ 01100001\ 11100011,\\
\boldsymbol{f} - 2 &= 10010010\ 10010010\ 01100001\ 01100001,\\
\boldsymbol{f} + 2 &= 01111000\ 01111000\ 11100011\ 11100011,\\
\boldsymbol{f} - \boldsymbol{u} &= 11000011\ 11000011\ 11110000\ 11110000,\\
S_2^{\vee} \boldsymbol{f} &= 11111010\ 11111010\ 11100011\ 11100011,\\
S_2^{\oplus} \boldsymbol{f} &= 11101010\ 11101010\ 10000010\ 10000010,\\
S_{\boldsymbol{u}}^{\oplus} \boldsymbol{f} &= 00111111\ 00111111\ 11000011\ 11000011.
\end{aligned}$$

The check of a triad is performed at first on initial rows of fragments constituting together the initial coefficient f^- of decomposition of function $f(\boldsymbol{x})$ by variables a and b. The rows, orthogonal to the initial row, are marked with value 1 in a computed vector $\boldsymbol{g}$ and are checked further on compatibility (non-orthogonality). With this purpose the couple of vectors $\boldsymbol{h}^0$ and $\boldsymbol{h}^1$ is evaluated. The same as initial vectors $\boldsymbol{f}^0$ and $\boldsymbol{f}^1$, they are Boolean vectors with 2^n components.

$\boldsymbol{h}^0 := (\boldsymbol{f}^0 - a) - b$	Obtaining of initial coefficient f^-
$\boldsymbol{h}^1 := (\boldsymbol{f}^1 - a) - b$	
$\boldsymbol{g} := S_c^{\vee} (\boldsymbol{h}^0 \boldsymbol{f}^1 \vee \boldsymbol{h}^1 \boldsymbol{f}^0)$	Selection of rows, orthogonal to the initial one
$\boldsymbol{h}^0 := S_a^{\vee} (S_b^{\vee} (\boldsymbol{f}^0 \boldsymbol{g})$	Checking them on compatibility
$\boldsymbol{h}^1 := S_a^{\vee} (S_b^{\vee} (\boldsymbol{f}^1 \boldsymbol{g})$	

If it appears, that $\boldsymbol{h}^0 \boldsymbol{h}^1 \neq \boldsymbol{0}$, the triad admits unsuitable. Otherwise the last rows of the fragment are checked, which constitute the final coefficient $\boldsymbol{f}^+$ (by that the symbol "–" in first two strings of the algorithm is changed for "+"). If it appears, that the triad all the same is unsuitable, other triads are tested, until a suitable one will not be found.

Search of the partition by a trace. If the considered triad (*a*, *b*)/*c* has appeared suitable, it is possible to assume, that it is a trace of the required partition. In this case the latter can be found, moving by the track, i. e. using the value of vector ***g*** obtained at the previous stage, and sequentially extending the sets ***u*** and ***v*** with initial values ***u*** = (*a*, *b*) and ***v*** = (*c*).

Let's begin from the set ***v***. Sorting out sequentially all elements *s* from the set $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$, we shall discover among them such ones, at which inclusion in set ***v*** the partition ***u***/***v*** remains suitable. With this purpose three operations are fulfilled for each element *s* :

$$\boldsymbol{e} := S_s^{\vee} \boldsymbol{g}$$
$$\boldsymbol{h}^0 := S_u^{\vee} (\boldsymbol{f}^0 \boldsymbol{e})$$
$$\boldsymbol{h}^1 := S_u^{\vee} (\boldsymbol{f}^1 \boldsymbol{e})$$

And if $\boldsymbol{h}^0 \boldsymbol{h}^1 = \boldsymbol{0}$, the element *s* is included into ***v***, which is implemented by the operations

$$\boldsymbol{v} := \boldsymbol{v} \cup \{s\}, \quad \boldsymbol{g} := \boldsymbol{e}.$$

Then the maximum extension of the set ***u*** is looked for. If it is known, that the required partition is disjoint, it is possible to put $\boldsymbol{u} = \boldsymbol{x} \setminus \boldsymbol{v}$. Otherwise, it is necessary to test all variables from the current value of the set $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$ and, if it is possible, to include them in set ***u***.

Check of the current element *s* we shall perform by an heuristic algorithm which operations are limited. It considers the initial coefficient f^- of decomposition of the function *f* by the current value of set ***u***, discloses orthogonal to it coefficients, checks them for compatibility and, in case of compatibility, includes element *s* into set ***u*** without further checking.

$$\boldsymbol{e} := \boldsymbol{u} \cup \{s\}$$
$$\boldsymbol{h}^0 := \boldsymbol{f}^0 - \boldsymbol{e}$$
$$\boldsymbol{h}^1 := \boldsymbol{f}^1 - \boldsymbol{e}$$
$$\boldsymbol{g} := S_v^{\vee} (\boldsymbol{h}^0 \boldsymbol{f}^1 \vee \boldsymbol{h}^1 \boldsymbol{f}^0)$$
$$\boldsymbol{h}^0 := S_u^{\vee} (\boldsymbol{f}^0 \boldsymbol{g})$$
$$\boldsymbol{h}^1 := S_u^{\vee} (\boldsymbol{f}^1 \boldsymbol{g})$$

If $\boldsymbol{h}^0 \boldsymbol{h}^1 = \boldsymbol{0}$, element *s* is included into ***u***, that is implemented by the operation ***u***: = ***e***.

So the set ***u*** and, therefore, the required partition ***u***/***v*** as a whole are found.

Experimental system

For the estimation of efficiency of the designed heuristic program of decomposition of Boolean functions and delimitation of its practical application an experimental system of generation of random examples and their solutions were designed.

The generation of examples is controlled by the following data-ins:

n – the number of arguments of the generated function *f*(***x***),

p and *q* – the numbers of variables in sets ***u*** and ***v*** of partition ***u***/***v*** on the set of variables ***x***,

r – the degree of uncertainty of the function *f*(***x***),

g – the number of an example, i. e. the serial number of the unit of a quasi-random sequence used at generation of the current example.

The type of partition is determined also: D (disjunctive) or ND (non-disjunctive).

The results of solution of the regarded examples are fixed by values of the following output parameters:

Nt – the number of surveyed triads when searching suitable ones,

Tc, Tt, Tv, Tu and Tw – time (in seconds), expended accordingly on preparation of an example, on finding of a suitable triad, on the extension of set ***v***, on the extension of set ***u*** and on solution of an example as a whole (excluding Tc).

Besides the quality of the solution is evaluated and it is decided, whether the calculated partition ***u***/***v*** and functions $h(\boldsymbol{u}, \boldsymbol{w})$ and $g(x, \boldsymbol{w}, \boldsymbol{v})$ coincide with given ones.

For simplification of the experimental research of the program a special mode is introduced into the system, at which a series of examples is generated and solved with coincident values of all parameters except one, for which a difference between adjacent values and their number are defined.

Experimental estimations of the program efficiency and the boundary of its practical application

A series of carried out experiments displays, that the circumscribed heuristic program operates safely enough, discovering exact solutions, if $8 < n < 28$ and $r < 30$. The errors can arise outside this range and, maybe, on the boundaries. The results of solution of some concrete examples and conclusions following from them are shown below.

Estimation of the upper bound by the number of variables n. Let's cite the results of an experiment, in which the number of variables n varies from 22 up to 28, and the values of remaining parameters are fixed.

Type	n	p	q	r	g	Nt	Tc	Tt	Tv	Tu	Tw	Result
D	22	11	11	20	1	1	0.34	0.16	1.56	0.00	2.08	OK
D	23	12	11	20	1	1	0.69	0.33	3.33	0.00	4.45	OK
D	24	12	12	20	1	5	1.47	1.90	7.20	0.00	10.79	OK
D	25	13	12	20	1	2	3.12	1.99	15.19	0.00	20.70	OK
D	26	13	13	20	1	2	6.18	4.04	34.07	0.00	45.20	OK
D	27	14	13	20	1	2	12.52	8.43	74.24	0.00	99.01	OK
D	28	14	14	20	1	2	24.97	31.22	1938.61	0.00	2096.32	OK

The results of the experiment are positive (OK – a retrieved partition coincides with the initial one) and display, that at given parameters the program discovers solution rather fast (in limits of a minute), except for the case $n = 28$, when the length of vector ***f*** becomes too big for the used RAM, that results in sharp increase of time Tv. They show also, that at $n < 28$ the time of solving grows twice at each increase of the number of variables n by unity. Let's remark, that Tu = 0, as an example of the task with disjoint partition is considered.

Estimation of the time of searching for a suitable triad. The results of the previous experiment display that at major p and q this time (Tt) is relatively small. The position varies at small values of parameters p and q. In that case, the number Nt of triads which are looked through in searches of a suitable one strongly increases and, as the corollary, grows the time of solution of the task as a whole, which is almost completely spent for this search.

Type	n	p	q	r	g	Nt	Tc	Tt	Tv	Tu	Tw	Result
ND	14	2	1	0	1	491	0.00	0.04	0.00	0.00	0.04	OK
ND	15	2	1	0	1	603	0.00	0.09	0.00	0.00	0.09	OK
ND	16	2	1	0	1	1250	0.01	0.36	0.01	0.00	0.37	OK
ND	17	2	1	0	1	1618	0.00	0.93	0.00	0.02	0.95	OK
ND	18	2	1	0	1	2055	0.02	2.37	0.01	0.05	2.43	OK
ND	19	2	1	0	1	2978	0.04	9.82	0.05	0.11	9.98	OK
……												
ND	27	2	2	0	1	2649	10.28	6812.28	48.64	95.09	6956.09	OK

Let's remark, that the value Nt is characterized by a major dispersion. For example, for a series of 12 random examples at fixed values of parameters (n, p, q, r) = (16, 2, 1, 0) the following results are obtained:

Nt =	1250	596	2799	498	2315	1598	4834	3797	2362	1637	3990	700
Tt =	0.36	0.18	0.80	0.14	0.66	0.46	1.39	1.08	0.67	0.47	1.15	0.21

With the purpose of specification of these data an experiment above ten random examples with parameters (Type, n, p, q) = (D, 24, 12, 12) was carried out. At r = 28 all ten examples were solved correctly, at r = 29 exact solutions were obtained only for seven examples, and at r = 30 all solutions have appeared false.

Estimation of the upper bound by the degree of uncertainty r. With growth of the degree of uncertainty of a Boolean function the probability of obtaining erratic solution at its decomposition grows also. The experiment over a series of random Boolean functions obtained as a result of composition at a strong partition with equal (whenever possible) values of parameters p and q was carried out. The maximum degree of uncertainty r_{max} of the function of n variables was defined, at which there was an exact solution. The results are shown in the following table:

n =	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
r_{max} =	0	1	6	8	14	19	19	21	22	23	25	25	26	26	28	27	29	29	29	29	30	30

Estimation of the lower bound by the number of variables n. At small number of variables many from suitable triads are not true traces of the partition, as it follows from the following experimental data received at a strong partition with equal (whenever possible) values of parameters p and q and following values of other parameters: n = 5, 6, … 16 and r = 15.

Number of variables	5	6	7	8	9	10	11	12	13	14	15	16
Total number of triads	30	60	105	168	252	360	495	660	858	1092	1365	1680
Number of suitable triads	28	44	80	78	54	90	77	90	126	147	196	224
Number of partition traces	6	9	18	24	40	50	75	90	126	147	196	224

The following table displays, how frequently it leads to a false solution. In conducted experiments series of 20 random examples with fixed values of parameters Type, n, p, q, r were solved and for each series the number of false solutions was counted up.

Type	n	p	q	r	Falsh	Type	n	p	q	r	Falsh
D	4	2	2	0	62	ND	4	2	2	0	62
D	5	3	2	0	76	ND	5	2	2	0	82
D	6	3	3	0	64	ND	6	2	2	0	55
D	7	4	3	0	50	ND	7	3	2	0	21
D	8	4	4	0	35	ND	8	3	3	0	6
D	9	5	4	0	6	ND	9	3	3	0	0
D	10	5	4	0	0	ND	10	4	3	0	0

*About finding functions h(**u**, **w**) and g(x, **w**, **v**)*. The main objective of decomposition of a Boolean function is finding a suitable partition ***u***/***v***, which permits to reduce the number of entry poles in blocks of the composition $g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v})$. Some interest represents finding functions $h(\boldsymbol{u}, \boldsymbol{w})$ and $g(x, \boldsymbol{w}, \boldsymbol{v})$. It is obvious, that, having found these functions, we uniquely determinate also the partition ***u***/***v***. However, from finding the partition it does not follow, that we have found also these functions.

That is confirmed by the following experimental data, received by solution of the task of decomposition on hundred random examples with parameters (Type, n, p, q) = (D, 10, 5, 5) and r taking values from 0 up to 26 including. The number of guessed right partitions is designated as Nr, the number of guessed functions – as Nf.

r	0	1	…	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Nr	100	98	…	98	98	98	97	95	92	86	75	63	47	28	16	13	10	6	6	3	1	0
Nf	100	98	…	98	97	97	96	94	88	82	64	52	33	15	7	4	2	1	1	0	0	0

Conclusion

An original heuristic algorithm of sequential two-block decomposition of partial Boolean functions of n variables on weak partitions on the set of arguments is implemented by a program and experimentally researched. It is shown, that the implementing program operates safely enough, discovering exact solutions if $8 < n < 28$ and if the function is defined not less than on 29/32 parts of Boolean space. The results of more detailed research of the program on boundaries of this range and outside it are cited.

Acknowledgement

The work was partially supported by Belarus Republican Fund of Fundamental Research {Project Ф07МС-034).

Bibliography

1. Perkowski M. A., Grigiel. A survey of literature on function decomposition, Version IV. November 20, 1995. Portland State University.

2. Zakrevskij A.D. Sequential decomposition of a Boolean function – the search for an appropriate partition on the set of arguments. - Reports of NAS of Belarus, 2007, v. 51, No 1, pp. 7-11 (in Russian).

3. Zakrevskij A.D. Parallel operations over neighbors in Boolean space. - Proceedings of the Sixth International Conference CAD DD-07, - Minsk 2007, vol. 2, pp. 6-13.

4. Zakrevskij A.D. Implementation of random events with a given probability. - Transactions of SFTI, Tomsk, 1965. - Is. 47, pp. 56-59 (in Russian).

5. Arkadij Zakrevskij. Decomposition of Boolean functions – recognizing a good solution by traces. – International Journal “Information Theories and Applications”, vol. 14, 2007, pp. 359-365.

Author information

Arkadij Zakrevskij - *United Institute of Informatics Problems of the NAS of Belarus, Surganov Str. 6, 220012 Minsk, Belarus; e-mail: zakr@newman.bas-net.by*

Nikolai R. Toropov - *United Institute of Informatics Problems of the NAS of Belarus, Surganov Str. 6, 220012 Minsk, Belarus; e-mail: toropov@newman.bas-net.by*

EXTENDED ALGORITHM FOR TRANSLATION OF MSC-DIAGRAMS INTO PETRI NETS

Sergii Kryvyy, Oleksiy Chugayenko

***Abstract**: The article presents an algorithm for translation the system, described by MSC document into Petri Net modulo strong bisimulation. Obtained net can be later used for determining various systems' properties. Example of correction error in original system with using if described algorithm presented.*

***Keywords**: MSC, Petri Net, model checking, verification, RAD.*

***ACM Classification Keywords**: D.2.4 Software/Program Verification - Formal methods, Model checking*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

The growing topicality of modern software systems and the rapidity imposed by pressing time-to-market demands for new approaches to the development process of high-performance and low-cost software. Namely, design productivity should be improved by means of new methodologies implementation due to the increase of the complexity of the systems. Formal methods begin to play a crucial role during design process. The use of formal methods during the initial stages of the development process can help to improve the quality of the later software, even if formal methods are not used in subsequent phases of development.

Protocol design is one of the most critical problems in distributed communication systems. Effective design methodology for protocol design requires formal models which are able to capture the inherent aspects of a system specification and verification tools that allow the designer to verify that a system satisfies its specification, to check the correctness of the system specification, and quickly explore alternative solutions.

Backgrounds

MSC is a modeling technique that uses a graphical interface, which was standardized by ITU (International Telecommunication Union) [1], [2]. It is usually applied to applications of the telecommunication domain, since they have properties of distributed reactive real-time systems. MSC diagrams are widely used in the early design stages of the systems development to capture system requirements. So, MSC is extremely suitable to capture the scenarios that a designer might want the system to exhibit (or avoid). MSC describes message flow between the instances, which present asynchronously communicating objects of the system or system entities like blocks, services or processes of the system. One MSC diagram describes a certain portion of system behavior or a scenario of communication between the instances. The set of the scenarios (MSC–diagrams) are captured as requirements which constitute a complete behavioral description of the system. Let us describe briefly the most basic MSC-constructs.

Instances and Messages. Instance is a basic primitive of MSC, which in graphics is presented as vertical line with its name. Message transmissions, which are acts of communication between instances, are presented by horizontal arrows with possible curve or tilt under angle for reflecting “overtaking” or “intersection” of messages. The beginning of the vector marks a sending of the message and its ending marks receiving of the message. Evens of sending and receiving of the messages are ordered along the instances so that sending of the message always happens earlier than its receiving. There is one more rule in standard MSC’2000 [1] for ordering events along the instances: everything located above happens earlier than that located below (except **coregion** part, where events are not ordered).

Creation and termination of instances may be specified within MSCs. An instance may be created by another instance. No message events before the creation can be attached to the created instance. The instance stop is the counterpart to the instance creation, except that an instance can only stop itself whereas an instance is created by another instance.

Conditions. Condition is used as for restricting or defining a set of MSC traces through indicating states of the system so for defining the composition of one MSC diagram from the several MSCs. Namely, the standard MSC'2000 [1] defines conditions of two types: setting condition and guarding condition.

Conditions of the first type are those which describe the current global system state (global condition), or some non-global state (nonglobal condition). In the latter case the condition may be local, i.e. attached to just one instance. Instances presenting dynamic objects can be began and finished, so far globality of a state considers dynamically changing set of instances.

Conditions of the second type restrict behavior of the MSC to execution of events in a certain part of MSC depending on the value of the given guarding condition.

Besides of composition role, conditions according to the standard [1] are the means of events synchronization. For example, if two instances share one and the same condition, then for each message between these instance its sending and receiving events shall happen both before or both after setting of the condition. If two conditions are ordered directly sharing the common instance, or indirectly through conditions on other instances, then this order must be respected on all instances that share these two conditions.

General Ordering. General ordering is used to impose additional orderings upon events that are not defined by the normal ordering given by the MSC semantics. For example, it may be used to specify that an event on one instance must happen before an otherwise unrelated event on another instance.

There cannot be both upwards and downwards steps on the same ordering. This means that it may consist of consecutive vertical and horizontal segments. The textual grammar defines the partial order relations by the keywords before and after. They indicate directly in what order the involved events must come in the legal traces.

Inlines. Inline expressions used to create various composition of events inside MSC diagram. They allow creating of alternative, parallel, loop compositions and exceptional and optional regions. Last two are special cases of alternative composition.

Environment and Gates. The gates represent the interface between the MSC and its environment. Any message or order relation attached to the MSC frame constitutes a gate. Due to possibility of MSC nesting, gates can be defined for MSC diagrams, MSC diagram references (reference expressions) and for inlines. For gate connections the associated gate definition must correspond with the actual gate. Message gates are used for message events and order gates are used for causal ordering.

Order gates represent uncompleted order relations where an event inside the MSC will be ordered relative to an event in the environment. Order gates are always explicitly named. Order gates are considered to have direction - from the event ordered first to the event coming after it. Also order gates are used on references to MSC diagrams (MSCs) in other MSCs. The actual order gates of the MSC reference are connected to other order gates or to events.

Message gates define the connection points of messages with the environment. The message gates are used when references to the MSC are put in a wider context in another MSC. A message gate always has a name. The name can be defined explicitly by a name associated with the gate on the frame. The actual message gates on the MSC reference are then connected to other message gates or instances. Similar to gate definitions, actual gates may have explicit or implicit names.

MSC Semantics. MSC is a language with formally defined semantics, which is based on the process algebra. Applying this formal semantics to MSC a process term can be derived for each MSC and each MSC specification. MSC language semantics based on process algebra was defined first for textual representation of MSC diagrams in the form of expressions of process algebra that was called denotation semantics. Operational semantics is defined via transitional rules added to algebraic expressions.

Petri Net. Ordinary Petri net is used as a formal model to define the semantics of MSC system and support analysis [3], [4].

Definition 1. A net is a triple $N = (P,T,F)$, such that P and T are disjoint sets of places and transitions respectively, and $F \subseteq (P \times T) \cup (T \times P)$ is binary incidence relation between places and transitions (flow relation).

On the basis of incidence relation F (flow relation) characteristic function $\overline{F} : (P \times T) \cup (T \times P) \to N$ is introduced, where N is the set of natural numbers.

The sets ${}^\bullet x = \{y \mid yFx\}$ and $x^\bullet = \{y \mid xFy\}$ denote the pre- and post set of arbitrary element $x \in P \cup T$ of net.

The following three conditions are required for the net $N = (P,T,F)$:

C1) $P \cap T = 0$,

C2) $(F \neq 0) \wedge [(\forall x \in P \cup T)(\exists y \in P \cup T) : xFy \vee yFx]$,

C3) $(\forall p_1, p_2 \in P)(p_1 = p_2 \wedge p^1 = p^2 \Rightarrow p_1 = p_2)$.

Definition 2. A marking of the net $N = (P,T,F)$ is the function $\mu : P \to N$. The equation $\mu(p) = k \in N$ means that place $p \in P$ has k tokens.

Definition 3. Petri net (PN) is a triple (N,μ_0,W), comprising a certain net N, certain initial marking μ_0, and W is weight function (or multiplicity of an arc).

A transition $t \in T$ is enabled at a marking μ of PN (N,μ_0) iff $\forall p \in {}^\bullet t : \mu(p) \geq \overline{F}(p,t)$. Such a transition can be fired, leading to the marking μ' according the following rule: $\forall p \in P : \mu'(p) = \mu(p) - \overline{F}(p,t) + \overline{F}(t,p)$. A sequence of transitions $\sigma = t_1,t_2,...,t_n$ is an occurrence sequence of a PN iff there exist marking $\mu_0,\mu_1,...,\mu$ such that $\mu_0 \vec{t_1} \mu_1 \vec{t_2} ... \vec{t_n} \mu$. It is said that μ is reachable from μ_0 by the occurrence of σ. The marking obtained by enabled sequences are said to be reachable.

A PN is said to be safe if any reachable marking has at most one token at each place.

A PN is said to be ordinary if all of its arc weights are 1's.

Definition 4. Marked Petri net is a pair (N,Σ), where N — Petri net and $\Sigma : T \to A$ — markup function over the alphabet A. If Σ is a partial function, unmarked transition are called "λ-transitions" and marked by the "empty symbol" λ.

Algorithm of translation of MSC Document to Petri Net

MSC subset used by algorithm. Presented algorithm processes the MSC 2000 constructions set with the following limitations:

1. Time constraints are ignored.
2. Timers are processed only as separate events.
3. MSC references are allowed only in HMSC.
4. MSC reference expressions are not processed.
5. Loop boundaries are ignored as treated as <1,inf> (or <0,inf> by user's choice).
6. Sequential MSC diagram connection is treated as strong sequence (i.e. all events in the first diagram shall be finished before the second one will start).

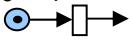
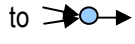
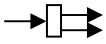
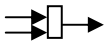
Algorithm assumes that source MSC document is syntactically and statically correct. Relation between MSC diagrams shall be given explicitly by HMSC.

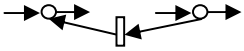
Algorithm description

Stage1 (Building of Trace Graph). During the first stage of the algorithm the Trace Graph will be build. This graph represents the traces, which a structurally possible in original MSC document (save loop iterations, which will be added during the next stage). Trace Graph creation consists of the following steps:

1. HMSC start construction translates to «start» node.
2. HMSC end construction translates to «end» node.
3. HMSC points of alternative branching translates to «alt-in» nodes.
4. HMSC points of branch joining translates to «alt-out» nodes.
5. HMSC par frame translates to «par-in» and «par-out» pair of nodes.
6. HMSC lines translates to edges between corresponding nodes.
7. MSC references in HMSC translates according the following rules (8 — 19).
8. For each MSC diagram created the nodes pair «msc-in» and «msc-out». This pair forms new synchronization zone. All edges, which corresponds to HMSC lines, drawn to this MSC reference, connect to the «msc-in» node. All edges, which corresponds to HMSC lines, drawn from this MSC reference, connect to the «msc-out» node.
9. Nodes, which corresponds to instance create events of MSC diagram, connect with «msc-in» node.
10. Nodes, which corresponds to instance end events of MSC diagram, connect with «msc-out» node.
11. Inline of «exc» and «opt» types translates as corresponding «alt» inlines.
12. Inline of «alt» type translates to «alt-in» and «alt-out» pair of nodes. Each of alternatives in this inline forms new synchronization zone.
13. Inline of «part» type translates to «par-in» and «par-out» pair of nodes. Each of alternatives in this inline forms new synchronization zone.
14. Inline of «loop» type translates to «loop-in» and «loop-out» pair of nodes. Inline body forms new synchronization zone.
15. All edges, which represent MSC lines, drawn to inline, connect to the corresponding «-in» nodes. All edges, which represent HMSC lines, drawn from inline, connect to the corresponding «-out» nodes.
16. Coregions are treated as par inline for all it's events.
17. All other valid MSC events translates to graph nodes; invalid events are skipped.
18. Edges which represent the order, explicitly shows in MSC diagram, added to graph.
19. Gates between MSC diagrams and between inlines resolved and new edges, which corresponds to gated messages and gated order relation, are added to graph.
20. In each synchronization zone edges, which do not correspond to domination relation, are removed. (Nested synchronization zone is treated as one node for outer zone.)

Stage2 (Building of Petri Net). On the second stage Trace Graph is used for resulting Petri Net building. Petri Net creation consists of the following steps:

1. Node «start» translated to  construction. Transition is marked by λ.
2. Node «end» translated to  construction. Transition is marked by λ.
3. Node «alt-in» translated to  construction. Number of out edges corresponds to number of alternatives.
4. Node «alt-out» translated to  construction. Number of in edges corresponds to number of alternatives.
5. Node «par-in» translated to  construction. Number of out edges corresponds to number of alternatives. Transition is marked marks by λ.
6. Node «par-out» translated to  construction. Number of in edges corresponds to number of alternatives. Transition is marked by λ.
7. «msc-in» and «msc-out» nodes are translated to transitions and marked by λ.

8. «loop-in» and «loop-out» pair translated to  construction. Transition is marked by λ. (If user choose <0,inf> loop boundaries, additional pass-through transition with λ mark added.)
9. Rest of graph nodes are translated to transitions marked by names of source MSC document elements.
10. Edges of the Trace Graph are translated to edges in Petri Net, which connects the corresponding nodes. If such edge connects transition with transition, addition place inserted in this edge.

Example

Let's show the algorithm work and usage on simple example. One of the simplest producer-consumer models as shown on the next MSC document with one MSC and one HMSC diagram (figures 1 and 2).

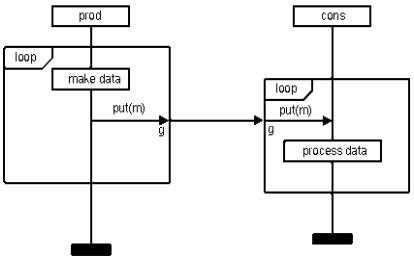


Fig. 1.

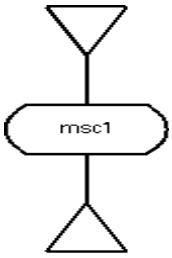


Fig. 2.

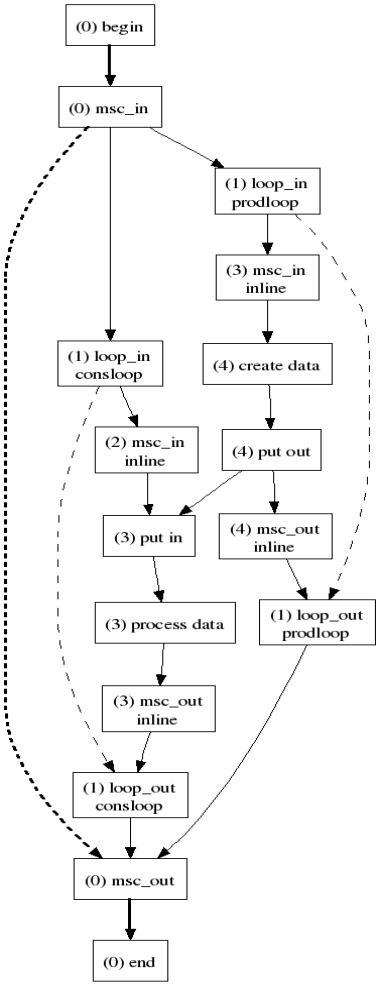


Fig. 3.

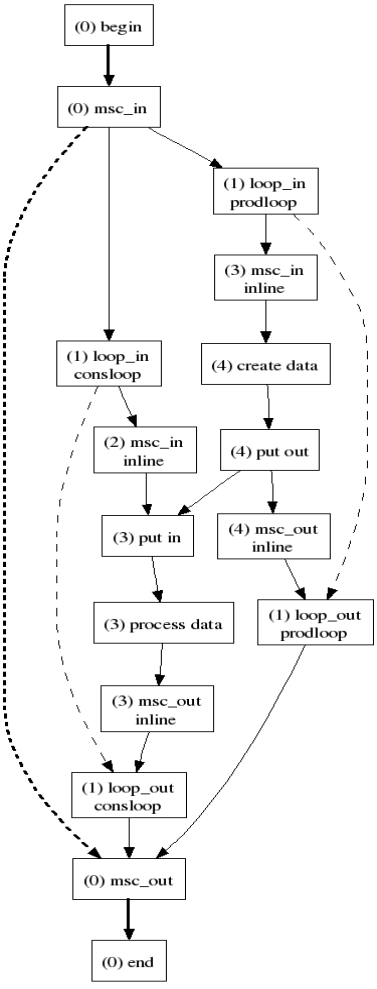


Fig. 4.

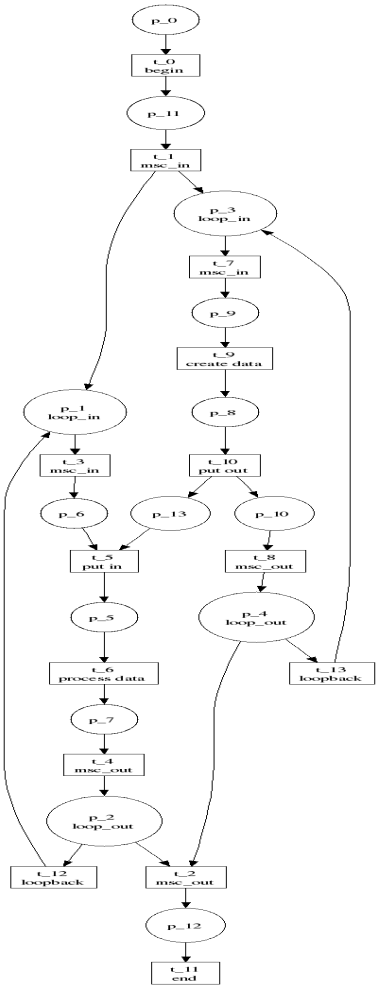


Fig. 5.

This diagram has two instances — producent (prod) and consument (cons). Producent infinitely produces some date (action «make data» in loop) and sends it to consument by message put(m) through gate g. Consument also has a loop, which allows it to infinitely process data («process data» action). After performing steps 1 — 19 we obtain the graph, shown on figure 3 and, after removing the non-dominated edges (step 20), on figure 4.

Then, after performing the Step 2 of algorithm, we obtain PN, shown on figure 5.

Now, let's try to analyze our system. First of all check liveness; to do this, calculate incidence matrix of obtained PN and find it's T-invariants. Incidence matrics is:

```
-1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 -1 0 0 0 0 0 0 0 0 1 0
0 0 -1 0 1 0 0 0 0 0 0 0 -1 0
0 1 0 0 0 0 0 -1 0 0 0 0 0 1
0 0 -1 0 0 0 0 0 1 0 0 0 0 -1
0 0 0 0 0 1 -1 0 0 0 0 0 0 0
0 0 0 1 0 -1 0 0 0 0 0 0 0 0
0 0 0 0 -1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 -1 0 0 0
0 0 0 0 0 0 0 1 0 -1 0 0 0 0
0 0 0 0 0 0 0 0 -1 0 1 0 0 0
1 -1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 -1 0 0
0 0 0 0 0 -1 0 0 0 0 1 0 0 0
```

And truncated solution set of the equation Ax=0 is (0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1), which shows, that transitions in our PN are alive except t_0 (begin MSC document), t_1 (begin MSC diagram), t_2 (begin MSC diagram) and t_11 (end MSC document) — just as expected. So, all events in the source can be executed infinitely. Then, check for PN's boundness. To perform this check, reduce PN as shown in [5] and create it's covering tree (figures 6,7).

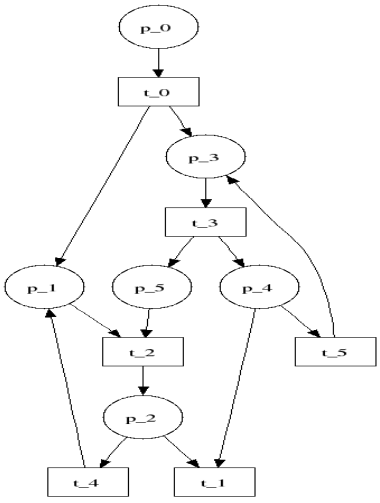


Fig.6.

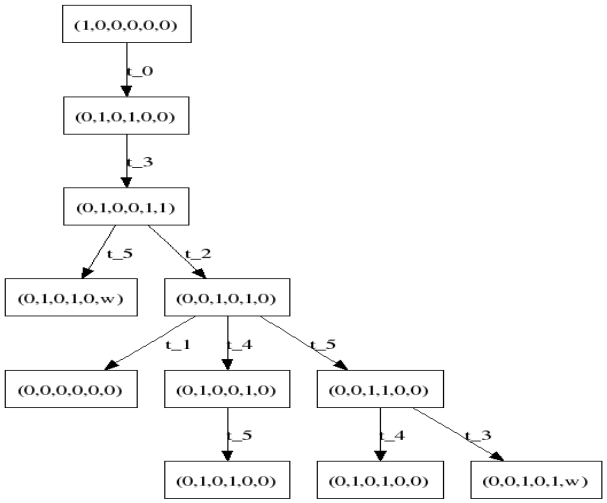


Fig. 7.

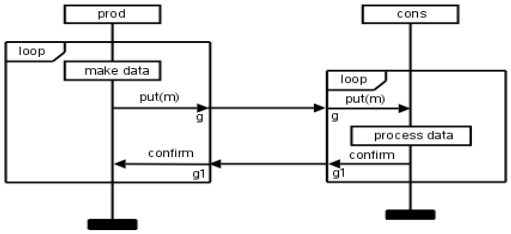


Fig. 8.

As we can see on covering tree, place p_5 of reduced PN is unbounded. It corresponds to place p_13 of original PN and means, that infinite number of messages put(m) can be stacked in original MSC (it can happens if consumer process data slower than producer creates it). To fix this, let's change original MSC by adding synchronization between consumer and producer (figure 8).

Now let's see on reduced PN and covering tree for corrected document (other steps have been omitted to save the article space).

Now we can see (figure 10) that our PN is bounded, so synchronization was enough to fix the found problem.

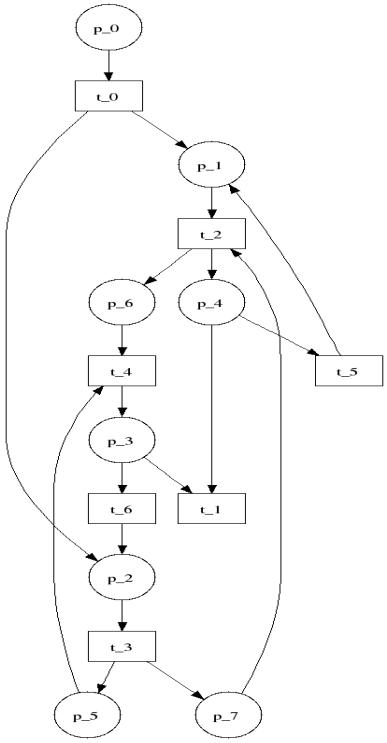


Fig. 9.

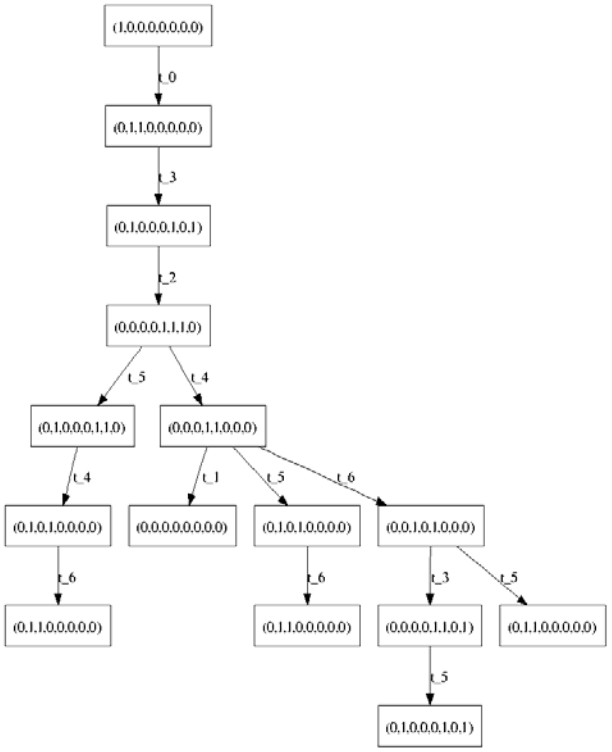


Fig. 10.

Conclusion

For conclusion we note, that the most significant feature of the given algorithm is its increased usability. The input set of MSC elements has been extended. Most of the existing design approaches require decomposing from the beginning the overall functionality into components. Extended input set of MSC elements in the given version of the translation algorithm makes it possible to apply different composing/decomposing techniques for subsequent PNs analysis.

The algorithm of translation MSC diagrams into Petri net, presented in the paper, considers only the subset of MSC language, therefore full implementation of reference MSC expressions and processing time constraints are our further research direction.

References

[1] ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU_TS, Geneva (2000).

[2] ITU-TS Recommendation Z.120 Annex B: Message Sequence Chart Annex B: Formal semantics of Message Sequence Charts. ITU_TS, Geneva (2001).

[3] Kryvyy, S., Matveyeva, L.: Algorithm of Translation of MSC-specified System into Petri Net. Fundamenta Informaticae, Vol.79 (2007), 1–15.

[4] Kryvyy, S., Matveyeva, L., Chugayenko O.: Extension of Algorithm of Translation of MSC specified system into Petri Net. Proc. of the CS&P'2007, 2007 – v2 – p.376—388

[5] Murata T. Petri Nets: Properties, Analysis and Verification. Proc. of the IEEE, 1989 — 77 — N4, p.65—74

Authors' Information

Kryvyi Sergii *– Glushkov Institute of Cybernetics NAS Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua*

Chugayenko Oleksiy *– Institute of Cybernetics NAS of Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: avch@avch.org.ua*

MINIMIZATION OF REACTIVE PROBABILISTIC AUTOMATA

Olga Siedlecka

***Abstract**: The problem of finite automata minimization is important for software and hardware designing. Different types of automata are used for modeling systems or machines with finite number of states. The limitation of number of states gives savings in resources and time. In this article we show specific type of probabilistic automata: the reactive probabilistic finite automata with accepting states (in brief the reactive probabilistic automata), and definitions of languages accepted by it. We present definition of bisimulation relation for automata's states and define relation of indistinguishableness of automata states, on base of which we could effectuate automata minimization. Next we present detailed algorithm reactive probabilistic automata's minimization with determination of its complexity and analyse example solved with help of this algorithm.*

***Keywords**: minimization algorithm, reactive probabilistic automata, equivalence of states of automata, bisimulation relation.*

***ACM Classification Keywords**: F. Theory of Computation, F.1 Computation by Abstract Devices, F.1.1 Models of Computation, Automata; F.4 Mathematical logic and formal languages, F.4.3 Formal Languages*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

The problem of finite automata minimization appeared in the end of fifties of last century and its main point is to find automata with the minimum number of states accepting the same language as input automata. During last fifty years many algorithms for minimization of finite deterministic automata came into existence, most of which (except Brzozowsky algorithm which is based on derivatives [Brzozowski, 1962]), is based on equivalence of states. One of the most popular minimization algorithms is Hopcroft and Ullman's algorithm with running time $O(|\Sigma|n^2)$ (where $|\Sigma|$ is the number of symbols in the alphabet, n is the number of states) [Hopcroft, 2000]. Another algorithm with the same time complexity, but better memory complexity ($O(|\Sigma|n)$) is Aho-Sethi-Ullman's algorithm [Aho, 2006]. The most efficient deterministic finite automata minimization algorithm is Hopcroft's algorithm [Hopcroft, 1971] with time complexity $O(|\Sigma|n\log n)$.

In the same period of time scientists were searching for another models of computation. They developed probabilistic automata [Rabin, 1963], which are extensions of Markov chains with read symbols [Sokolova, 2004], models of finite automata over infinite words [Thomas, 1990], timed automata [Alur, 1994], hybrid automata [Henzinger, 1998] etc. We can find their ontological review in article: [Kryvyi, 2007]. It became important to find minimization algorithms for new types of automata. So far minimization of reactive probabilistic automata hasn't been described.

Probabilistic automata

It exists many types of probabilistic automata which differs with properties, applications or probability distributions (continuous or discrete). Their review we can find in article [Sokolova, 2004]. Hereunder we itemize few of probabilistic automata's types with discrete probability distribution: the reactive automata, the generative automata, the I\O automata, the Vardi automata, the alternating model of Hansson, the Segala automata, the bundle probabilistic automata, the Pnueli-Zuck automata and others.

The algorithm showed in article was formulated for the reactive probabilistic automata.

A reactive probabilistic automata is a triple $PA=(Q, \Sigma, \delta)$, where Q is the finite set of states, Σ is the finite set of input symbols (an alphabet), δ is the transition probability function given by $\delta: Q\times\Sigma\to D(Q)$ (where $D(Q)$ is the set of all discrete probability distribution on the set Q) [Sokolova, 2004] .

An initial reactive probabilistic automata with accepting states is a five $PA=(Q, \Sigma, \delta, q_0, F)$, in which we have additionally two elements: q_0 - a member of Q, is the start state, $F\subset Q$ is the set of final (accepting) states.

After reading given symbol automata is in state of superposition of states: $p_0q_0+p_1q_1+...+p_nq_n$, where $p_0+p_1+...+p_n=1$. Henceforth we will use shorter name of probabilistic automata within the meaning of initial reactive probabilistic automata with accepting states. An example of this type of automata we show on figure 1.

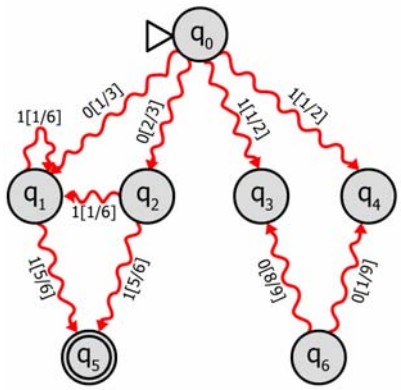


Fig.1. The initial reactive probabilistic automata with accepting states

The probability of going from state q_1 to state q_2 after reading symbol σ we denote as $\delta(q_1,\sigma)(q_2)=p$. An extended transition probability function, denoted by the same notation δ is given by:

$$\delta(q_1, w\sigma)=\sum_{q\in Q}\delta(q_1, w)(q)\cdot\delta(q,\sigma)$$ [Cao, 2006].

The language accepted by the probabilistic automata is defined as function $L_{PA}:\Sigma^*\to[0,1]$, such that:

$$\forall w\in\Sigma^*, L_{PA}(w)=\sum_{q\in F}\delta(q_0,w)(w)$$ [Cao, 2006].

We say that language L is recognized with bounded error by an automata PA with interval (p_1,p_1), if $p_1<p_2$ and $p_1=sup\{P_w|w\notin L\}$, $p_1=inf\{P_w|w\in L\}$ [Golovkins, 2002].

We say that language L is recognized with probability p, if the language is recognized with interval $(1-p,p)$ [Golovkins, 2002].

We say that language L is recognized with probability $1-\varepsilon$, if for every $\varepsilon>0$ there exist an automata which recognizes the language with interval $(\varepsilon_1,1-\varepsilon_2)$, where $\varepsilon_1,\varepsilon_2\leq\varepsilon$ [Golovkins, 2002].

Bisimulation and indistinguishableness

Let R be an equivalence relation on the set S, and let $P_1,P_2\in D(S)$ be discrete probability distributions. Then $P_1\equiv_R P_2\Leftrightarrow\forall C\in S/R: P_1[C]=P_2[C]$, where C is an equivalence class [Sokolova, 2004].

Let R be an equivalence relation on the set S, let A be a set, and $P_1,P_2\in D(S)$ be discrete probability distributions. Then:

$P_1\equiv_{R,A}P_2\Leftrightarrow\forall C\in S/R,\ \forall a\in A: P_1[a,C]=P_2[a,C]$ [Sokolova, 2004].

Let $PA_1=(S, \Sigma, \delta)$ and $PA_2=(T, \Sigma, \delta)$ be two reactive probabilistic automatas. A bisimulation relation $R\subseteq S\times T$ exists if for all $(s,t)\in R$ and for all $\sigma\in\Sigma$:

- if $\delta(s,\sigma)=P_1$ then there exists a distribution P_2 with $t\in T$ such that $\delta(t,\sigma)=P_2$ and $P_1\equiv_{R,\Sigma}P_2$ [Sokolova, 2004].

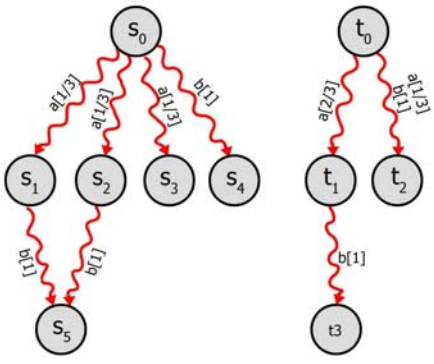


Fig.2. The bisimulation relation on PA

States $(s,t)\in R$ we call bisimilar, what is denoted by $s\approx t$.

Let $PA_1=(S, \Sigma, \delta, q_0, F_S)$ and $PA_2=(T, \Sigma, \delta, q_0, F_T)$ be two initial reactive probabilistic automata with accepting states. We can define indistinguishableness relation $N\subseteq S\times T$, if for all $(s,t)\in N$ and for all $\sigma\in\Sigma$:

- $(s,t)\in N^0$ if and only if $((s\in F_S\wedge t\in F_T)\vee(s\notin F_S\wedge t\notin F_T))$,

- $(s,t) \in N^k$ if and only if $(s,t) \in N^{k-1}$ and
  - if $\delta(s,\sigma)=P_1$ then exists the probability distribution P_2 with $t \in T$ such that $\delta(t,\sigma)=P_2$ and $P_1 \equiv_{R,\Sigma} P_2$.

For $n=|Q|$, we have $N \subseteq N^{n-2} \subseteq N^{n-3} \subseteq ... \subseteq N^1 \subseteq N^0$. States s,t we call indistinguishable, what is denoted by $s \equiv t$, if there exists indistinguishableness relation N, such that $(s,t) \in N$.

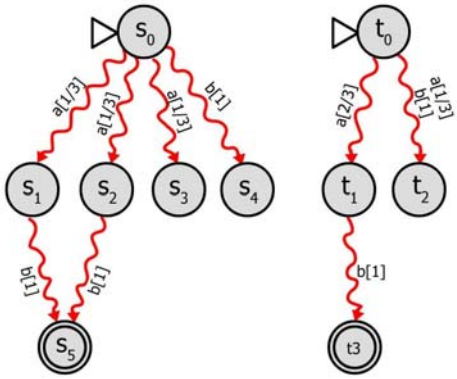


Fig.3. The indistinguishableness relation on PA

Minimization of reactive probabilistic automata

A probabilistic automata $PA=(Q, \Sigma, \delta, q_0, F)$ recognizing language L with probability p we call minimal, if there doesn't exist automata with smaller number of states recognizing language L with not smaller probability.

A minimization of probabilistic automata parts on two steps:

- elimination of unreachable states (probability to reach those states is 0),
- joining of indistinguishable states (using indistinguishableness relation).

First we show on below code elimination of unreachable states:

Alg.1. Algorithm of elimination of unreachable states:

```
INPUT: PA=(Q,Σ,δ,q0,F)- reactive probabilistic automata.
OUTPUT: PA'=(Q',Σ,δ',q0,F') - reactive probabilistic automata without unreachable
states, recognizing the same language as PA.
1.    FOR ALL {q∈Q} DO
2.      markedStates[q]←0;
3.    END FOR
4.      S.push(q0); markedStates[q]←1; pr←0;
5.    WHILE {S≠∅} DO
6.      p←S.first();
7.      S.pop();
8.      FOR ALL {σ∈Σ} DO
9.        FOR ALL {q∈Q} DO
10.         pr←δ(p,σ)(q);
11.         IF {pr≠0 ∧ markedStates[q0]=0} THEN
12.           S.push(q);
13.           markedStates[q]←1;
14.         END IF
15.       END FOR
16.     END FOR
17.   END WHILE
18.   FOR ALL {q∈Q} DO
19.     IF {markedStates[q]=1} THEN
20.       Q'.push(q);
21.     END IF
22.   END FOR
23.   F'←F∩Q;
24.   FOR ALL {q∈Q} DO
25.     IF {markedStates[q]=1} THEN
26.       FOR ALL {p∈Q} DO
27.         IF {markedStates[p]=1} THEN
28.           FOR ALL {σ∈Σ} DO
29.             δ'(q,σ)(p)←δ(q,σ)(p);
30.           END FOR
31.         END IF
32.       END FOR
33.     END IF
34.   END FOR
```

In this algorithm S is auxiliary stack, on which we put states, which we can reach with non-zero probability going out from the start state q_0. The transition probability function $\delta(p,\sigma)(q)$ gives probability pr of reaching state q, going out from state p, reading symbol σ. The running time of the algorithm time is bounded by:

$$T(n,|\Sigma|) \leq a(7+9n+2|\Sigma|n+2n^2+6|\Sigma|n^2)+c(4+8n+2|\Sigma|n+3n^2+5|\Sigma|n^2),$$

where a is time of an assignment and c is time of comparison, clearly $O(|\Sigma|n^2)$ is the time complexity of this algorithm.

In the algorithm of joining indistinguishable states we use already defined indistinguishableness relation. In one word, states to be indistinguishable, have to be in the same equivalence class, and must have the same probability distribution for symbols and equivalence classes, which can be reach from this states. Inspired by Hopcroft-Ullman's algorithm [Hopcroft, 2000], first we assume that all pairs of states are indistinguishable, above that, that first element of pair is member of final states' set and second isn't. Next analysing all pair of states and all symbols we find distinguishable states, until the moment that any change is made. Algorithm analyses probability distributions of reaching state from state.

Alg.2. Algorithm of joining indistinguishable states:

```
INPUT: PA=(Q,Σ,δ,q0,F) - reactive probabilistic automata.
OUTPUT: PA'=(Q',Σ,δ',q0',F') - minimal reactive probabilistic automata recognizing
language L_PA.
1.    FOR {i←0; i<|Q|; i←i+1} DO
2.      FOR {j←0; j≤i; j←j+1} DO
3.        IF {(qi∈F ∧ qj∉F) ∨ (qi∉F ∧ qj∈F)} THEN
4.          Dqi,qj←1;
5.        ELSE
6.          Dqi,qj←0;
7.        END IF
8.      END FOR
9.    END FOR
10.   FOR {i←1; i<|Q|; i←i+1} DO
11.     FOR {j←0; j<i; j←j+1} DO
12.       IF {Dqi,qj=0} THEN
13.         FOR ALL {σ∈Σ} DO
14.           E1←0, E2←0, N1←0, N2←0;
15.           FOR ALL {p∈Q} DO
16.             IF {Dqi,p=0} THEN
17.               E1←E1+δ(qi,σ)(p);
18.             ELSE
19.               N1←N1+δ(qi,σ)(p);
20.             END IF
21.             IF {Dqj,p=0} THEN
22.               E2←E2+δ(qj,σ)(p);
23.             ELSE
24.               N2←N2+δ(qj,σ)(p);
25.             END IF
26.           END FOR
27.           IF {E1≠E2 ∨ N1≠N2} THEN
28.             Dqi,qj←1; break;
29.           END IF
30.         END FOR
31.       END IF
32.     END FOR
33.   END FOR
34.   Q'←Q, F'←F, q0'←q0;
35.   FOR {i←1; i<|Q|; i←i+1} DO
36.     FOR {j←0; j<i; j←j+1} DO
37.       IF {Dqi,qj=0} THEN
38.         Q'←Q'\{qi,qj}, Q'←Q'∪{qij};
39.         IF{qi∈F} THEN
40.           F'←F'\{qi,qj}, F'←F'∪{qij};
```

```
41.          END IF
42.          IF {j=0} THEN
43.            q₀←qⱼ;
44.          END IF
45.        END IF
46.      END FOR
47.    END FOR
48.    FOR ALL {q₁×q₂×σ ∈ Q′×Q′×Σ} DO
49.      IF {q₁∉Q ∧ q₁=p₁p₂ : p₁p₂∈Q} THEN
50.        δ′(q₁,σ)(q₂)← δ(p₁,σ)(q₂);
51.      ELSIF {q₂∉Q ∧ q₂=p₁p₂ : p₁p₂∈Q } THEN
52.        δ′(q₁,σ)(q₂)← δ(q₁,σ)(p₁)+δ(q₂,σ)(p₂);
53.      ELSE
54.        δ′(q₁,σ)(q₂)← δ(q₁,σ)(q₂);
55.      END IF
56.    END FOR
```

Analyzing algorithm in details: on input we have reactive probabilistic automata; on output we get minimal automata that accept the same language as input automata. In lines 1 to 9 we tentatively fill structure *D*, which is lower triangular matrix of all combination of automata's states. In place where one of the states is final and second isn't, we set value 1, because states are distinguishable. In other case we set 0, providing that all other pairs of states are indistinguishable. In lines 10 to 33 is the main part of algorithm, which decides if states are equal or not, comparing probability distributions. First (line 12) we verify if pair of states is indistinguishable $D_{qi,qj}=0$ (otherwise it makes no sense in analyzing them). For every symbol from alphabet Σ we reset value of auxiliary variables *E1, E2, N1, N2*, in which we will sum probabilities of reaching distinguishable states *N* or indistinguishable states *E*. States will be generally recognized as indistinguishable if values of *E1, E2* and *N1, N2* will be respectively equal. If for two analyzed states, for any symbol of alphabet, we get different values of those variable, loop is interrupted (line 28), because states are distinguishable and we go to next iteration. In the last part of algorithm (from line 34) we create output automata, so we replace indistinguishable states by single states, and calculate values for transition probability function (from line 48). Depending, if we analyze reaching state or going out from new state, values of probability will be summed or copied. The running time of the algorithm is bounded by:

$$T(n,|\Sigma|) \leq a(5 + 4.5n - 3.5|\Sigma|n + 7.5n^2 + 2|\Sigma|n^2 + 3n^3 + 1.5|\Sigma|n^3) +$$
$$c(2 + 7n - 2.5|\Sigma|\, n + 7n^2 + |\Sigma|\, n^2 + 7n^3 + 1.5|\Sigma|n^3),$$

so complexity will be $O(|\Sigma|n^3)$.

Lets analyze steps of both algorithms on example from figure 1. First we reset table *markedStates[q_i]*, which size is 7 (automata has 7 states). We push on stack start state. Next we mark with *1* field for this state in table *markedStates[q_0]*. We pop from the stack start state and push those, which we can reach from start state reading symbol *0*, with nonzero probability (those will be q_1, q_2) and for symbol *1*, respectively q_3, q_4, in every case marking them with *1* in table *markedStates[q_i]*. In next iteration we search for states we can reach from states put on the stack. Finally, the only state, which wasn't marked is q_6. In next steps we exclude it from the set of states of automata.

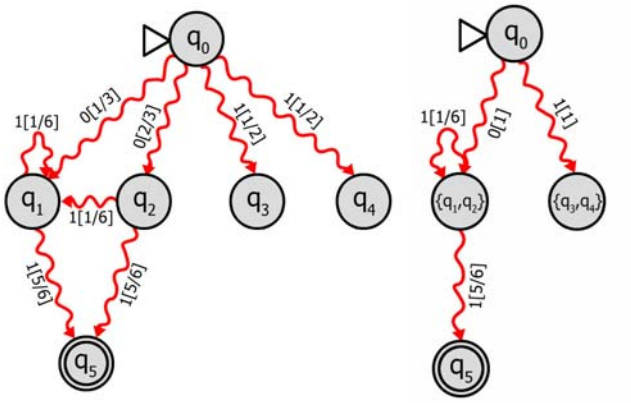


Fig.4. a) Elimination of unreachable states b) Joining of indistinguishable states

The algorithm of joining indistinguishable states in first part fill structure $D_{qi,qj}$ with *1* in those places where one of states is final, and second isn't – for all combinations of other states with state q_5. Next we check successively all

combinations of states and sum probabilities of going out from this states in variables *E1, E2, N1, N2*, for example for states q_1, q_0, values for this variables are *E1=0, E2=1, N1=0, N2=0*, so this pair of states is distinguishable and $D_{qi,qj}=1$. Finally structure $D_{qi,qj}$ has value *1* only for pairs: q_1, q_2 and q_3, q_4, which will be replaced by new single states q_{12}, q_{34}. Probabilities for reaching those states will be summed, and for going out from them will be copied.

Conclusion

In article we define indistinguishableness relation for reactive probabilistic automata, what give us opportunity to build minimization algorithm, with complexity $O(|\Sigma|n^3)$. Algorithms will terminate, because number of states or symbols in alphabet is always limitation for iterations (and we work on finite sets). The probability for accepting words doesn't change because it is respectively summed or copied.

The definition of indistinguishableness relation and minimization algorithm is the base for further work on adequate algorithm for quantum automata.

Bibliography

[Aho, 2006] A.V. Aho, M.S. Lam, R. Sethi, J.D. Ullman, Compilers: Principles, Techniques, and Tools (2nd Edition). Addison Wesley, 2006.

[Alur, 1994] R. Alur, D.L. Dill, A theory of timed automata. Theoretical Computer Science 126, 2 (1994), pp. 183 - 235.

[Brzozowski, 1962] J. A. Brzozowski, Canonical regular expressions and minimal state graphs for definite events. In Proceedings of the Symposium on Mathematical Theory of Automata (1962), vol. 12 of MRI Symposia Series, Polytechnic Press of the Polytechnic Institute of Brooklyn, pp. 529 - 561.

[Cao, 2006] Y. Cao, L. Xia, M. Ying, Probabilistic automata for computing with words. ArXiv Computer Science e-prints (2006).

[Golovkins, 2002] M. Golovkins, M. Kravtsev, Probabilistic reversible automata and quantum automata. Lecture Notes In Computer Science 2387 (2002), p. 574.

[Henzinger, 1998] T.A. Henzinger, P.W. Kopke, A. Puri P.Varaiya, What s decidable about hybrid automata? Journal of Computer and System Sciences 57 (1998), pp. 94 - 124.

[Hopcroft, 1971] J.E. Hopcroft, An n log n algorithm for minimizing the states in a finite automaton. In The Theory of Machines and Computations, Z. Kohavi, Ed. Academic Press, 1971, pp. 189 - 196.

[Hopcroft, 2000] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation (2nd Edition). Addison Wesley, 2000.

[Kryvyi, 2007] S. Kryvyi, L. Matveeva, E. Lukianova, O. Siedlecka, The Ontology-based view on automata theory. In Proceedings of 13-th International Conference KDS-2007 (Knowledge-Dialog-Solution) (Sofia, 2007), ITHEA, Ed., vol. 2, pp. 427-436.

[Rabin, 1963] M.O. Rabin, Probabilistic automata. Information and Controle 6 (1963), pp. 230 - 245.

[Sokolova, 2004] A. Sokolova, E. de Vink, Probabilistic automata: System types, parallel composition and comparison. In Validation of Stochastic Systems: A Guide to Current Research (2004), LNCS 2925, pp. 1 – 43.

[Thomas, 1990] W. Thomas, Automata on infinite objects. Handbook of theoretical computer science: formal models and semantics B (1990), pp. 133 - 191.

Authors' Information

Siedlecka Olga *– Institute of Computer and Information Sciences, Czestochowa University of Technology, ul. Dabrowskiego 73 42-200 Czestochowa, Poland; e-mail: olga.siedlecka@icis.pcz.pl*

PARALLELIZATION OF LOGICAL INFERENCE FOR CONFLUENT RULE-BASED SYSTEM[1]

Irene Artemieva, Michael Tyutyunnik

***Abstract**: The article describes the research aimed at working out a program system for multiprocessor computers. The system is based on the confluent declarative production system. The article defines some schemes of parallel logical inference and conditions affecting scheme choice.*

***Keywords**: Logical Inference, parallel rule-based systems*

***ACM Classification Keywords**: D 3.2 – Constraint and logic languages, I 2.5 Expert system tools and techniques.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

Production systems (or rule-based systems) are used to develop knowledge-based systems [1]. The set of rules describes the problem-solving method of the system application. Production systems are preferable to algorithmic languages as rules usually require terminology of the application domain and are assigned by its ontology [2] which allows to get a problem-solving method understood by the user.

In confluent production systems, the output does not depend on the order of rules of the logical inference. It means that all the rules are independent of each other, i.e. the confluent production language does not need extra language constructions for writing parallel programs and thus for constructing parallel systems to solve application tasks in knowledge-based systems. This differs them from other classes of production systems and parallel programming systems based on logical languages [3-5].

A language processor – a production language compiler allocates computations according to a process. When generating an object code, a language processor analyzes characteristics of the source program, a set of constraints imposed by the computing environment, characteristics of input data defined by the user and selects the most applicable scheme of the parallel logical inference.

The aim of this article is to describe schemes of the parallel logical inference implemented by the language processor of the confluent production system and conditions affecting the scheme choice.

Production System Language Characteristics

The research on ontologies and design knowledge-based systems on their basis makes it possible to formulate requirements for the language of the confluent production system.

1. The language must allow to represent a problem-solving method as a set of solving methods for subtasks described by the modules. Each module must have its interface, i.e. data description, that can be used by another module or that are required for its operation/performance. There must be an explicit condition of a module call, i.e. among the rules there must be rules the right part of which is a module call.

2. The language must allow to use operations on numeric data and sets. The language must allow to use limited logical and mathematical quantifiers that are analogs of loops in the rules.

[1] This paper was made according to the program № 14 of fundamental scientific research of the Presidium of the Russian Academy of Sciences, the project 06-I-П14-052

3. The language must admit rules that are dependent on parameters (rule scheme). The scheme assigns a set of rules, i.e. it can be considered as an analog of a subprogram in the algorithmic language.

The language admits rules of two kinds:

(1) (prefix) $P(X) \rightarrow S_1(X_1)\&\ldots\&S_k(X_k)$, where $P(X)$ is a formula, $S_1(X_1)$, …, $S_k(X_k)$ are simple formulas, X, X_1, …, X_k are vectors of terms];

(2) (prefix) $P(X) \rightarrow NameMod(S_1, \ldots, S_k)$ is a rule for module call, where $P(X)$ is a formula, $NameMod(S_1, \ldots, S_k)$ is a module name, X is vector of terms, and S_1, …, S_k are arguments of the module.

The formula before the symbol «→» is a production antecedent, the formula after the symbol «→» is its consequent. The antecedent is a logical expression made up of relations, functional terms, atomic formulas, generalized formulas according the following rules.

Each rule must meet the main antedecent for variables: $V(\{S_1(X_1), \ldots, S_n(X_n)\}) \cup V(P(X)) \subseteq V(prefix)$, where $V(O)$ is a set of variables included into O (O can be any construction).

Prefix is a sequence of descriptions of variables $(v_1{:}t_1)(v_2{:}t_2)\ldots(v_m{:}t_m)$ where $(v_i{:}t_i)$ is a description of a variable, v_i is a variable, t_i is a term for all i=1,...,m. Term t_1 does not contain free variables. For i=2,..., m only variables v_1, v_2,...,v_{i-1} can be free variables of term t_i. The sequence of descriptions can be empty. All variables v_1, v_2,..., v_m are dually different.

The following construction can be called a scheme:

(3) $NameSch([.CONST]w_1, [.CONST]w_2, \ldots, [.CONST]w_n)$: rule where

NameSch is a scheme name, w_1, w_2, …, w_n are variables, rule is of the kind (1). Variables w_1, w_2, …, w_n are formal parameters of the scheme. The scheme body is a rule and must contain formal parameters. «[.CONST]» means that «.CONST» can be absent.

The following construction can be called a scheme concretization:

(4) $NameSch(zw_1, zw_2, \ldots, zw_n)$ where NameSch is a scheme name, zw_1, zw_2, …, zw_n are terms that are actual parameters of the scheme.

The scheme is an analog of a procedure in programming languages, the scheme concretization is an analog of a procedure call.

A domain symbol (a term of the domain ontology) – name n; variable v; sets I, R, S; empty set $\varnothing$; set $\{t_1,t_2,\ldots,t_k\}$ where $t_1,t_2,\ldots,t_k$ are terms; intervals $I[t_1,t_2]$, $R[t_1,t_2]$ where t_1 and t_2 are terms; expression t_1 ¤ t_2 where t_1 and t_2 are terms, sign «¤» $\in \{+, -, *, /\}$, t_1 ¤ t_2 where t_1 and t_2 are terms-sets, sign «¤» $\in \{\cup, \cap, \setminus\}$ is a sign of operation on sets; $\mu(t)$ is a power of set where t is a term-set, $(Z_n\,(v : t_1)\ t_2)$ is a quantifier term where $Z_n \in \{+, *, \cup, \cap\}$, v is a variable (index of a quantifier term), t_1 is a term-set that assigns a range of v, t_2 is a term (the body of a quantifier term) that contains operation index v; $f(t_1,\ldots,t_k)$ is a functional term where f is a functional symbol (a term of the domain ontology), $t_1,\ldots,t_k$ are terms (arguments of functional term) are terms;

$p(t_1,\ldots,t_k)$ or $\neg p(t_1,\ldots,t_k)$ where $p(t_1,\ldots,t_k)$ is an atomic formula, p is a predicate symbol (a term of the domain ontology), $t_1,\ldots,t_k$ are terms (arguments of atomic formula); t_1 @ t_2 where t_1 and t_2 are terms, sign «@» is a sign of mathematical relation or relation on sets are simple formulas.

Logical expression f_1 @ f_2 where «@» $\in$ {&, V}; quantifier formula $(Z_n\,(v : t)\ f)$ where $Z_n \in$ {&, V}, v is a variable (index of a quantifier formula), t is a term-set that assigns a range of v, f I a formula (the body of a quantifier formula) that contains index v are formulas.

Information Graph Definition

A language processor is a compiler that translates a text in the production language into the object code in the algorithmic high-level language. The object code implements the logical inference assigned by the production rules and contains calls of modules of the run period support environment. Before the code generation the language processor makes the program information graph and analyzes its characteristics.

An aligned cyclic graph the vertices of which are rules and arcs of which indicate information relations between rules, i.e. arcs connect those rules that exchange data, is called the information graph. The arc between two vertices exists if the following antedecent is met: IF(π_j) $\cap$ THEN(π_i) $\neq \varnothing$ where IF(π_j) = {o_1', …, o_a'} – a set of terms of a domain included in the antedecent of the rule π_j, THEN(π_i) = {o_1'', …, o_b''} – a set of terms of a domain – arguments of the module called in the consequent of the rule π_i, or a set of terms of a domain included in the consequent of the rule π_l, $i \neq j$. The rule π_j will be called dependent on π_i. The information graph is created for each module. So, the information graph of a program is an array of information graphs of its modules.

Let us consider the structures used for representing the information graph of each module and different properties of a module that can be defined on basis of its graph. The element (i,j) of the incidence matrix IncMatrix with dimensions $\mu(\Pi_m) \times \mu(\Pi_m)$ where $\mu(\Pi_m)$ means the number of module rules is equal to 1 if the j-th rule is a direct child of the i-th rule, and is equal to 0 otherwise. LoopMatrix stores the information about the graph vertices that are included in a loop: elements correspondent to the rules not included in a loop are equal to 0, and included – 1. The element of an array iLoopAr with the number i is equal to -1, if the rule i is not included in any of the loops, is equal to 0 – the rule is included in one of loops but is not a loop entry; if the rule is a loop entry, the element is equal to the number of direct children of this rule. The number of direct children of the rule i is a value of each element with the number i of an array iParentsAr.

Using the Information Graph at the Parallelization of Logical Inference

This paper suggests using the "client-server" architecture when a separate process is a dispatcher (main process), other processes are handling processes (dependent processes) for constructing a parallel production system. The main process inputs and outputs data, synchronizes them and exchanges data with dependent processes; it prioritizes rules and provides each process with a subprogram to process a rule. Each dependent process executes a subprogram that implements the logical inference for the rule, i.e. it searches for all substitutions at which the condition of the rule applicability is true and for each substitution it performs actions defined by the consequent of the rule and passes the received data to other processes. Below there are schemes of the workflow of the main process and dependent process.

Before the main process starts to work, iCurParentsAr – a copy of iParentsAr – is created, at the same time if the information graph contains loops, we must change values of elements of the array iCurParentsAr in the following way: if iLoopAr[i] > 1, then iCurParentsAr[i] = iLoopAr[i], i.e. substitute rules-loop entries for the number of parents that do not belong to loops. Elements of the array iCurParentsAr change their values during the calculations. The array element i gets equal to -1 if the rule i is being processes, -2 – if the rule i has been processed.

Scheme 1a (main process):

Calculations Begin: $\mu(P_f) = \mu(P)$; $P_w = \varnothing$. Block 1.

LOOP: While exist iCurParentsAr[i] = 0 or $P_w \neq \varnothing$, do: Block 2; Block 3. Loop End.

Block 4. Calculations End.

Block 1 (*Start all rules which appropriate to root vertices*):

For every free process j from P_f do:

For every element i from array iCurParentsAr: If iCurParentsAr[i] = 0,

then Send(Q_i, i, j); iCurParentsAr[i] = -1; $P_f = P_f \setminus \{j\}$; $P_w = P_w \cup \{j\}$.

Block 1 End.

Here Send(Q_i, i, j) is a procedure that sends data set Q_i into process j and informs process j of the necessity to calculate rule i; $\mu(P)$ is the number of slave processes for calculations; $\mu(P_f)$ is the number of free processes. Data set Q_i contains all the values of the objects included in the condition of rule i.

Block 2 (*Receive and synchronize results*):

1. Recv(Z, i, j); $P_w = P_w \setminus \{j\}$; $P_f = P_f \cup \{j\}$.

2. iCurParentsAr[i] = -2.

3. For all vertices k such as

iCurParentsAr[k] > 0, do:

3.1. If IncMatrix[i,k] = 1,

then iCurParentsAr[k] = iCurParentsAr[k] – 1;

3.2. If iLooptAr[i] > 0 & iCurParentsAr[k] = -2 & iLoopAr[k] > 0 & THEN(i) $\cap$ IF(k) $\neq \varnothing$

then iChangedLoopAr[k] = 1.

4. Data = Data $\cup$ Z;

Block 2 End.

Here Recv(Z, i, j) is a procedure that receives from process j data set Z that are results of computing rule i. Data set Z contains values of the objects included in the consequent of rule i. Data is a data set that contains all the values of all the objects included in the rules.

Block 3 (*Assign rules ready for computing to free processes*):

For every free process j do:

For every element i from array iCurParentsAr:

If iCurParentsAr[i] = 0

then Send(Q_i, i, j); iCurParentsAr[i] = -1.

If $P_w = \varnothing$ & not exist elements k from array iCurParentsAr such as iCurParentsAr[k] = 0 then:

For every element t from array iCurParentsAr:

If iCurParentsAr[t] = -2 & iLoopAr[t] > 0 then:

For all vertices s:

If LoopMatrix[t,s]=1 & iCurParentsAr[s]>0

then iCurParentsAr[s] = 0;

Goto Block 3.

Block 3 End.

Block 4 (*Make rules which appropriate to loop vertices ready for repeated calculations*):

If exist elements i from array iChangedLoopAr such as iChangedLoopAr[i] = 1then:

For all rules do:

If k – (distant) child of rule i

then iChangedLoopAr[k] = 1.

For every element t from array iChangedLoopAr:

If iChangedLoopAr[t] = 1

then iCurParentsAr[t] = iParentsAr[t];

If iLoopAr[t]>1 then iCurParentsAr[t]=0.

Goto Block 3.

else:

Block 4 End.

Here iChangedLoopAr is an integer array where the element with number i is equal to 1 if loop rule i has been computed and then appear new values for the objects included in its antecedent; otherwise the element with number i is equal to 0. This structure is filled in the course of computing the rules and is used to construct a children list, the children must be computed again.

Scheme 1b (slave process):

Calculations Begin: wRecv(Z, i); wCalc(i, Z, Q); wSend(Q, i); Calculations End

Here wRecv(Z, i) is a procedure that receives from the main process data set Z that contains values of the objects included in the antedecent of rule i; wSend(Q, i) is a procedure that sends into the main process data set Q that are the results of computing rule i; wCalc(i, Z, Q) is a procedure that computes rule i with the help of the logical inference.

Tuple Passing at Incomplete Rule Computation

The previous scheme rigidly specifies that the dependent rule cannot be computed until the rules it is dependent on have been computed. This scheme does not have such a restriction – the next rule waits for at least one tuple – the result of the application of the rule it depends on but not the termination of all the rules-parents. If each next tuple appears at the beginning of the rule application, there can be a situation when all the rules are processed parallel regardless of how they are connected informationally. However, due to the restrictions connected with the number of processes of cluster computer free for computation, the number of rules that work parallel cannot exceed the number of free processes.

Let us complete the above schemes with a series of new operations that will allow loading free processes with those rules the only parents of which are being computed.

Let iParentsIdAr be an integer array the dimensions of which coincide with the number of module rules, the element with number i being equal to the number of the parent if vertex i has the only parent. SendPFrom(p_{from}, Q_i, i, j) is a procedure that sends data set Q in process j and informs process j of the necessity to calculate rule i, process j must receive from process p_{from} a set of next tuples for the objects included in the antecedent of rule i. Data set Q contains all the values of the objects included in the antecedent of rule i. SendPTo(p_{to}, p_i) is a procedure that sends in process p_i that applies rule i the message about the necessity to send to process p_{to} tuples for those objects included in the antecedent of rule processed by p_{to}.

Block X1 (*Assign additional rules to calculations using tuples passing*):
For every free process j from P_f do:
  For every element i from array iCurParentsAr:
    If iCurParentsAr[i] = -1 then
      For every element k from array iParentsIdAr:
        If iParentsIdAr[k] = i then
          iParentsIdAr[j] = -1;
          SendPFrom(p_{from}, Q_k, k, j);
          iCurParentsAr[k] = -1; $P_f = P_f \setminus \{j\}$; $P_w = P_w \cup \{j\}$.
          SendPTo(p_{to}, p_i);
Block X1 End.

Scheme 2a (main process):
Calculations Begin:
$\mu(P_f) = \mu(P)$; $P_w = \varnothing$.
Block 1.
Block X1.
LOOP: While exist iCurParentsAr[i] = 0 or $P_w \neq \varnothing$, do:
  Block 2.
  Block 3.
  Block X1.
LOOP End.
Block 4.
Calculations End.

Scheme 2b (slave process):
Calculations Begin:
flag = 0; $p_{to\_} = 0$;
wRecv_(p_{from}, Z, i);
If $p_{from} > 0$ then flag = 1;
Block 1.
If flag = 1 then:
  wRecvPFrom(p_{from}, Z_i);
  If $Z_i \neq \varnothing$ then $Z = Z \cup Z_i$;
  else flag = 0;
wCalc(i, Z, Q);
If wRecvPTo(p_{to}) = 1then $p_{to\_} = p_{to}$;
If $p_{to\_} > 0$ then wSend_($p_{to\_}$, Q);
Block 1 End.
If flag = 1 then Goto Block 1.
wSend(Q, i);
Calculations End.

Block X1 loads all the free processes with the rules that can receive tuples from their computed parents and informs the processes that compute parents of the necessity to pass tuples to other processes.

The scheme of the workflow of the dependent process is a modified scheme 1b. To describe it we use the following procedures.

wRecv_(p_{from}, Z, i) is a procedure that is an extended version of procedure wRecv. wRecv_ receives from the main process data set Z that contains the values of the objects included in the antecedent of rule i and receives the number of process p_{from} from where next tuples can come. If p_{from} = 0, tuples from other processes will not be sent. $Z \equiv \{O_1, \ldots, O_z\} \equiv \{\{k^1_1, \ldots, k^1_{k1}\}, \ldots, \{k^z_1, \ldots, k^z_{kz}\}\}$.

wRecvPFrom(p_{from}, Z_i) is a procedure that receives from slave process p_{from} data set Z that contains the value of the objects included in the antecedent of rule i. $Z_i \equiv \{O_1, \ldots, O_z\} \equiv \{\{k^1_1, \ldots, k^1_{k1}\}, \ldots, \{k^z_1, \ldots, k^z_{kz}\}\}$.

wRecvPTo(p_{to}) is a function that receives from the main process the number of slave process p_{to} to which it is necessary to send tuples. The function returns 0 if the number of the process from the main process has not been received, and it returns 1 if it has.

wSend_(p_{to}, Q) is a procedure that sends to slave process p_{to} data set Q that is the current result of the computed rule i. Data set Q contains all the values of the objects included in the consequent of rule i. $Q \equiv \{O_1, \ldots, O_q\} \equiv \{\{k^1_1, \ldots, k^1_{k1}\}, \ldots, \{k^q_1, \ldots, k^q_{kq}\}\}$.

Applying this scheme we can launch in parallel the process that will process the rule dependent on data not when the rule-parent has been performed, but when attributions for the objects found in the course of calculations start to come. Thus, if the dependence on data allows, one can launch all the rules in parallel as the correspondent attributions appear. This implies that the period of applying all the rules can be shorter than the period of applying the rules using the first scheme.

Conclusion

The above schemes use such characteristics of the information graph as the number of its vertices, the number of its branches that can be processed in parallel, etc. To choose a scheme of parallelization of the logical inference one has to know not only the characteristics of the graph but also architecture and system constraints imposed by the computing environment. There may be the following constraints:

1. The number of free system processes. If the number of the processes is more or equal to the number of the rules of the program, this case is convenient for calculations as one can specify in advance the particular rule for each process. If there are less processes than rules, then the rules are assigned to the processes dynamically.

2. The period of the rule application. In the course of computing a rule there may be a situation when this rule is processed many times longer than any other rule of the logical program. In this case there may be an idle time of the system that awaits the end of calculations of this rule. One of the solutions is to send intermediate results of the rule calculation to other process that process dependent rules.

3. The structure of the program information graph that is assigned by a set of information graphs of modules that are part of the program. The graph characteristics define the number of graph sections that can be executed in parallel. One has to analyze the graph to assign a rule for a free dependent process. The more branches are there in the graph, the stronger is the possibility of parallelizing calculations of rules. As one does not know the exact time of computing a rule, the maximum number of processes P_{opt} that can be performed in parallel with each other is defined in the following. For each vertex of the graph they build a set of vertices into which there are no ways from the given vertex and which are not parents (and far parent as well) of this given vertex. After computing the maximum from the number of the elements of all the sets, one will have the sought P_{opt}.

The closer is P_{opt} to the number of the rules in module $\mu(\Pi_m)$, the more efficient will be the parallelization of the task and quicker will be the calculations, i.e. $E = P_{opt} / \mu(\Pi_m)$ – tends to 1 (E defines the average fraction of the rule calculation by a separate process). From the definition P_{opt} it follows that P_{opt} will be bigger if the graph has more direct children for one parent, i.e. there is wide branchiness of the graph.

For each graph P_{opt} is invariable if the rules are executed completely: first – parents, then – children. However, one can start to pass intermediate results to the dependent rules without awaiting the completion of one rule. In this case if in the rule-parent there is at least one tuple of new values for the object included in the antecedent of the rule-child, the process that maintains the rule-parent sends the tuple to the process assigned to process the dependent rule. Doing this with all the rules and assigning a rule for a process, the system of the logical inference can assign all the rules for execution in parallel with each other. It implies that P_{opt} is always equal to the number of rules $\mu(\Pi_m)$ (and $\mu(P)$ must be equal to $\mu(\Pi_m)$) in module m, E = 1.

Hardware constraints in the form of relatively small number of free processes, memory allocation according to processes and temporary delays in the course of data passing between processes on the one hand and multilevel module calls on the other hand constrain module execution in parallel. Therefore sequential module execution can be considered optimal. In the course of the rule computation there can be a number of module calls, then the main process does not launch new rules for computing any more, completes the rest and then passes control to each called module one by one.

Bibliography

[1]. Gavrilova T.A., Khoroshevsky V.F. Intellectual System Knowledge Bases. (In Russian) – SPb.: Piter, 2000.

[2]. Kleshchev A.S., Artemjeva I.L. Mathematical models of domain ontologies // Int. Journal on Inf. Theories and Appl., 2007, vol 14, № 1. PP. 35-43.

[3]. M. Wallace, St. Novello, and J. Schimpf. ECLiPSe: A Platform for Constraint Logic Programming. Technical report, IC-Parc, Imperial College, London, 1997. http://citeseer.ist.psu.edu/wallace97eclipse.html http://citeseer.ist.psu.edu/update/38822

[4]. NESL - A Parallel Programming Language. http://www.cs.cmu.edu/~scandal/nesl.html

[5]. Boon S. Ang, Derek Chiou, Larry Rudolph and Arvind. The START-VOYAGER Parallel System. Massachusetts Institute of Technology, Laboratory for Computer Science. http://citeseer.ist.psu.edu/ang98startvoyager.html

Authors' Information

Irene L. Artemieva – *artemeva@jacp.dvo.ru*

Michael B. Tyutyunnik – *michaelhuman@gmail.com*

Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences;

5 Radio Street, Vladivostok, Russia

SEQUENCING JOBS WITH UNCERTAIN PROCESSING TIMES AND MINIMIZING THE WEIGHTED TOTAL FLOW TIME

Yuri Sotskov, Natalja Egorova

***Abstract**: We consider an uncertain version of the scheduling problem to sequence set of jobs J on a single machine with minimizing the weighted total flow time, provided that processing time of a job can take on any real value from the given closed interval. It is assumed that job processing time is unknown random variable before the actual occurrence of this time, where probability distribution of such a variable between the given lower and upper bounds is unknown before scheduling. We develop the dominance relations on a set of jobs J. The necessary and sufficient conditions for a job domination may be tested in polynomial time of the number $n = |J|$ of jobs. If there is no a domination within some subset of set J, heuristic procedure to minimize the weighted total flow time is used for sequencing the jobs from such a subset. The computational experiments for randomly generated single-machine scheduling problems with $n \leq 700$ show that the developed dominance relations are quite helpful in minimizing the weighted total flow time of n jobs with uncertain processing times.*

***Keywords**: Scheduling, robustness and sensitivity analysis.*

***ACM Classification Keywords**: F.2.2 Nonnumerical algorithms and problems: Sequencing and scheduling.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

There are scheduling problems in real life, where job processing times may be evaluated with high reliability before scheduling, and the vast majority of academic research assumes that job processing times are either deterministic (see book [Tanaev, Sotskov, Strusevich, 1994] and the first part of book [Pinedo, 1995]) or random variables with known probability distributions (the second part of [Pinedo, 1995]). However, it is not realistic to assume all the job processing times have known probability distribution for many other practical scheduling problems. For the most scheduling environments, job processing times are unknown variables and the only information that can be certainly obtained before scheduling is about lower and upper bounds for a job processing time. As such, a schedule obtained by assuming a certain probability distribution may not be close to the optimal schedule in practical realization of the process. Due to this reason, methods of construction of optimal and approximate schedules are practically important for scheduling problems with uncertain (interval) processing times [Kouvelis, Yu, 1997; Sotskov, Sotskova, 2004].

In this paper, we address a scheduling problem when it is impossible to obtain reliable probability distributions for the job processing times. Namely, it is assumed that the processing time of a job can take any real value from the given interval of uncertainty, regardless of the values taken by the processing times of other jobs. More precisely, we consider the non-preemptive single-machine sequencing problem with interval processing times to minimize the weighted sum of the job completion times.

The paper is organized as follows. In the second section, problem setting is given. The third section contains a literature review. The forth section reminds a known-result for a single-machine scheduling problem with the fixed processing times and the weighted total flow time criterion. The fifth section contains the necessary and sufficient condition over which a job dominates another one (i.e., for each set of possible processing times there exists an optimal permutation with the same order of these two jobs). An illustrative example is given in the sixth section. Computational results for randomly generated instances with interval processing times are given in the seventh section. The last section presents a brief conclusion.

Problem Setting

There are $n \geq 2$ jobs $J = \{J_1, J_2, ..., J_n\}$ to be processed on a single machine. For each job $J_i \in J$, positive weight $w_i > 0$ is given. Processing time p_i of a job $J_i \in J$ may take any real value between given lower bound $a_i \geq 0$ and upper bound b_i, $b_i \geq a_i$, which are only known before scheduling. Real number C_i is equal to the completion time of the job $J_i \in J$ and criterion $\sum w_i C_i = \sum_{i=1}^{n} w_i C_i$ denotes minimization of sum of the weighted completion times of n jobs. Let $S = \{\pi_1, \pi_2, ..., \pi_{n!}\}$ denote a set of all permutations $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n})$ of n jobs from the set $J = \{J_1, J_2, ..., J_n\}$. By adopting the three-field notation $\alpha | \beta | \gamma$ introduced in [Graham et al., 1976], we denote the scheduling problem of searching an optimal permutation within set S that minimizes the value $\sum_{i=1}^{n} w_i C_i$ as $1 | a_i \leq p_i \leq b_i | \sum w_i C_i$. A set $T = \{p : a_i \leq p_i \leq b_i, J_i \in J\}$ of vectors $p = (p_1, p_2, ..., p_n)$ of the processing times is a rectangular box in the space of non-negative n-dimensional real vectors. If a vector p of the processing times is known before scheduling (i.e., $a_i = b_i, J_i \in J$), then problem $1 | a_i \leq p_i \leq b_i | \sum w_i C_i$ becomes conventional problem $1 || \sum w_i C_i$ with the fixed job processing times. As it is proven in [Smith, 1956], the latter problem can be solved in $O(n \log_2 n)$ time. We call sequencing problem $1 | a_i \leq p_i \leq b_i | \sum w_i C_i$ an uncertain (sequencing) problem in contrast to problem $1 || \sum w_i C_i$, called a deterministic one.

Literature Review and Definition

In case of the uncertain problem $1 | a_i \leq p_i \leq b_i | \sum w_i C_i$, there may not exists a unique schedule that remains optimal for all possible realizations of the job processing times. Therefore, in [Daniels, Kouvelis, 1995], so-called robust schedule minimizing the worst-case absolute or relative deviation from optimality (called worst-case regret) was proposed to hedge against processing time uncertainty. In [Daniels, Kouvelis, 1995; Yang, Yu, 2002], uncertain problem $1 | a_i \leq p_i \leq b_i | \sum C_i$ with minimizing total flow time (i.e., it was assumed that $w_i = 1$ for each job $J_i \in J$) has been considered. In [Averbakh, 2000; Averbakh, 2001; Daniels, Kouvelis, 1995; Yang, Yu, 2002; Lebedev, Averbakh, 2006] along with *continuous* intervals of possible processing times defined by the above set T, the processing time uncertainty was described through a finite *discrete* set $T^*, |T^*| = h,$ of possible processing time vectors (called scenarios). Each scenario $p \in T^*$ represents fixed processing times for job set J, which can be realized with some positive (but unknown before scheduling) probability. For a specific scenario $p \in T^*$, deterministic problem $1 || \sum C_i$ arises which can be solved using optimal job permutation defined due to the following SPT rule: *Sort the jobs J according to non-decreasing order of their processing times*.

While deterministic problem $1 || \sum C_i$ is computationally simple, finding a permutation which minimizes the worst-case regret to the uncertain counterpart with discrete set of possible scenarios $T^*, |T^*| = h,$ is computationally hard problem. E.g., in [Daniels, Kouvelis, 1995], it was proven that to find a permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$ minimizing the worst-case absolute regret is binary NP-hard problem (see [Garey, Johnson, 1979] for definition) even for two possible scenarios: $h = 2$. In [Yang, Yu, 2002], it was proven that to find a permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$ minimizing the worst-case relative regret is binary NP-hard problem for two possible scenarios as well. In [Yang, Yu, 2002], it was proven that to find a permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$ minimizing the worst-case absolute (relative) regret is unary NP-hard problem [Garey, Johnson, 1979] for unbounded number h of possible scenarios.

Worst-case regret is also defined for the processing time uncertainty described through a rectangular box T of possible vectors p. In [Lebedev, Averbakh, 2006], it was proven that minimizing the worst-case absolute regret for

problem $1 \mid a_i \le p_i \le b_i \mid \sum C_i$ is binary NP-hard problem if intervals of the processing times have the same center for all jobs J. In [Averbakh, 2001], it was shown by an example that there is no direct relationship between the complexity of the uncertain problem with the given finite discrete set of possible scenarios $T^*, \mid T^* \mid = h,$ and the complexity of the uncertain problem with the given set T of n continuous intervals of possible scenarios.

Summarizing this overview, we can observe that for the most classical polynomially solvable deterministic scheduling problems, their uncertain counterparts with the worst-case regret criterion become binary or unary NP-hard problems. In fact, even for existence of only two scenarios of possible processing times (h=2), to minimize the absolute or relative regret implies a time-consuming search over set S of n! permutations of n jobs. In order to overcome this computational complexity in some special cases, we propose to use searching the minimal set of dominant schedules (permutations) introduced in [Lai, Sotskov, 1999] for solving the uncertain job-shop problem $J \mid a_i \le p_i \le b_i \mid C_{\max}$ with the makespan objective function: $C_{\max} = \max\{C_i : J_i \in J\}$.

Definition 1: *Set of permutations (schedules) S(T)* $\subseteq S$ *is a minimal dominant set for the uncertain problem* $\alpha \mid a_i \le p_i \le b_i \mid \gamma$, *if for any vector* $p \in T$ *set S(T) contains at least one permutation (schedule), which is optimal for the deterministic problem* $\alpha \mid\mid \gamma$ *with vector* p *of the job processing times provided that any proper subset of set S(T) loses such a property.*

A minimal dominant set *S(T)* was investigated in [Allahverdi, Sotskov, 2003; Allahverdi, Aldowaisan, Sotskov, 2003; Lai, Sotskov, 1999; Leshchenko, Sotskov, 2007] for the makespan criterion $C_{\max}$, and in [Allahverdi, Aldowaisan, Sotskov, 2003; Sotskov, Allahverdi, Lai, 2004] for the total flow time criterion $\sum C_i$. In particular, work of [Sotskov, Allahverdi, Lai, 2004] was addressed to the total flow time in a two-machine flow-shop with the interval processing times: $F2 \mid a_i \le p_i \le b_i \mid C_{\max}$. A geometrical algorithm has been developed for solving the flow-shop problem $Fm \mid n = 2, a_i \le p_i \le b_i \mid \sum C_i$ with m machines and two jobs. For uncertain flow-shop problems with two or three machines, sufficient conditions have been identified when the transposition of two jobs minimizes the total flow time. Work of [Allahverdi, Aldowaisan, Sotskov, 2003] was addressed to the case of separate setup times with the criterion $C_{\max}$ or $\sum C_i$. Namely, the processing times were fixed while each setup time was relaxed to be a distribution-free random variable within given lower and upper bounds. Dominance relations have been identified for an uncertain flow-shop problem with two machines. In [Allahverdi, Sotskov, 2003], for a two-machine flow-shop problem $F2 \mid a_i \le p_i \le b_i \mid C_{\max}$, sufficient conditions have been identified when the transposition of two jobs minimizes the makespan $C_{\max}$. In [Leshchenko, Sotskov, 2007], the necessary and sufficient conditions were used for the case when a single schedule dominates all the others, and the necessary and sufficient conditions were used for the case when it is possible to fix the optimal order of two jobs for the makespan criterion $C_{\max}$ with interval job processing times.

The formula for calculating stability radius of an optimal schedule (i.e., the largest value of independent variations of the job processing times for the schedule to remain optimal) has been provided in [Sotskov, Sotskova, Werner, 1997] for a job-shop problem $Jm \mid a_i \le p_i \le b_i \mid C_{\max}$ with m machines. Stability radius of an optimal schedule was investigated for problem $Jm \mid a_i \le p_i \le b_i \mid C_{\max}$ in [Lai, Sotskov, 1999; Sotskov, Wagelmans, Werner, 1998], and for problem $Jm \mid a_i \le p_i \le b_i \mid \sum C_i$ in [Brasel, Sotskov, Werner, 1996]. In contrast to references [Brasel, Sotskov, Werner, 1996; Lai, Sotskov, 1999; Sotskov, Sotskova, Werner, 1997; Sotskov, Wagelmans, Werner, 1998], where exponential algorithms based on exhausting enumeration of the semi-active schedules (see p. 284 in [Tanaev, Sotskov, Strusevich, 1998]) were derived for constructing minimal dominant set $S(T)$ for uncertain job-shop problems, in this paper, we show how to find set $S(T)$ for the problem $1 \mid a_i \le p_i \le b_i \mid \sum C_i$ in polynomial time. Next, we present an auxiliary result for the deterministic problem $1 \mid\mid \sum w_i C_i$.

Deterministic Sequencing Problem

In [Smith, 1956], it was proven that problem $1||\sum w_i C_i$ can be solved in $O(n\log_2 n)$ time due to the following sufficient condition for optimality of permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$:

$$\frac{w_{i_1}}{p_{i_1}} \geq \frac{w_{i_2}}{p_{i_2}} \geq ... \geq \frac{w_{i_n}}{p_{i_n}}. \quad (1)$$

It is easy to prove that inequalities (1) are also necessary conditions for optimality of permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$ for the problem $1||\sum w_i C_i$.

Theorem 1: *Permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_n}) \in S$ is optimal for the problem $1||\sum w_i C_i$ if and only if inequalities (1) hold.*

Proof: *Sufficiency* of condition (1) for optimality of permutation π_i was proven in [Smith, 1956].

Next, we prove *necessity* of condition (1) for optimality of permutation π_i by contradiction method.

Let permutation $\pi_i = (J_{i_1}, J_{i_2}, ..., J_{i_{r-1}}, J_{i_r}, J_{i_{r+1}}, J_{i_{r+2}}, ..., J_{i_n}) \in S$ be optimal for the problem $1||\sum w_i C_i$. However, for the latter permutation at least one inequality from condition (1) is violated, e.g., we assume that the following opposite inequality holds:

$$\frac{w_{i_r}}{p_{i_r}} < \frac{w_{i_{r+1}}}{p_{i_{r+1}}}, \quad (2)$$

where $r \in \{1,2,...,n-1\}$. Let us consider permutation $\pi'_i = (J_{i_1}, J_{i_2}, ..., J_{i_{r-1}}, J_{i_{r+1}}, J_{i_r}, J_{i_{r+2}}, ..., J_{i_n}) \in S$, which defers from permutation π_i by transposition of jobs J_{i_r} and $J_{i_{r+1}}$. We obtain the following equalities provided that notation $\Phi(\pi_i) = \Phi(J_{i_1}, J_{i_2}, ..., J_{i_n}) = \sum_{k=1}^{n} w_{i_k} C_{i_k}$ is used:

$$\Phi(\pi_i) = \sum_{q=1}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k}, \quad \Phi(\pi'_i) = \sum_{q=1}^{r-1} w_{i_q} \sum_{k=1}^{q} p_{i_k} + w_{i_{r+1}} (\sum_{k=1}^{r-1} p_{i_k} + p_{i_{r+1}}) + w_{i_r} \sum_{k=1}^{r-1} p_{i_k} + \sum_{q=r+2}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k}.$$

Let us calculate the difference of the objective function values defined for permutation π_i and permutation π'_i:

$$\Phi(\pi_i) - \Phi(\pi'_i) = \sum_{q=1}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k} - \left[\sum_{q=1}^{r-1} w_{i_q} \sum_{k=1}^{q} p_{i_k} + w_{i_{r+1}} \left(\sum_{k=1}^{r-1} p_{i_k} + p_{i_{r+1}} \right) + w_{i_r} \sum_{k=1}^{r+1} p_{i_k} + \sum_{k=r+2}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k} \right] =$$

$$= \sum_{q=1}^{r-1} w_{i_q} \sum_{k=1}^{q} p_{i_k} + w_{i_r} \sum_{k=1}^{r} p_{i_k} + w_{i_{r+1}} \sum_{k=1}^{r+1} p_{i_k} + \sum_{k=r+2}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k} -$$

$$- \left[\sum_{q=1}^{r-1} w_{i_q} \sum_{k=1}^{q} p_{i_k} + w_{i_{r+1}} \left(\sum_{k=1}^{r-1} p_{i_k} + p_{i_{r+1}} \right) + w_{i_r} \sum_{k=1}^{r+1} p_{i_k} + \sum_{k=r+2}^{n} w_{i_q} \sum_{k=1}^{q} p_{i_k} \right] =$$

$$= w_{i_r} \sum_{k=1}^{r} p_{i_k} + w_{i_{r+1}} \sum_{k=1}^{r+1} p_{i_k} - w_{i_{r+1}} \left(\sum_{k=1}^{r-1} p_{i_k} + p_{i_{r+1}} \right) - w_{i_r} \sum_{k=1}^{r+1} p_{i_k} =$$

$$= w_{i_r} \left(\sum_{k=1}^{r} p_{i_k} - \sum_{k=1}^{r+1} p_{i_k} \right) + w_{i_{r+1}} \left(\sum_{k=1}^{r+1} p_{i_k} - \sum_{k=1}^{r-1} p_{i_k} - p_{i_{r+1}} \right) = w_{i_r} (-p_{i_{r+1}}) + w_{i_{r+1}} p_{i_r} = w_{i_{r+1}} p_{i_r} - w_{i_r} p_{i_{r+1}}.$$

Thus, the following equality holds: $\Phi(\pi_i) - \Phi(\pi'_i) = w_{i_{r+1}} p_{i_r} - w_{i_r} p_{i_{r+1}}$. If we multiply both left-hand side and right-hand side of the latter inequality by factor $p_{i_r} p_{i_{r+1}}$, we obtain inequalities $w_{i_r} p_{i_{r+1}} < w_{i_{r+1}} p_{i_r}$ and $w_{i_{r+1}} p_{i_r} - w_{i_r} p_{i_{r+1}} > 0$ which implies: $\Phi(\pi'_i) < \Phi(\pi_i)$. The latter inequality contradicts to the above assumption

that permutation π_i is optimal for the problem $1||\sum w_iC_i$. The contradiction obtained implies the necessity of condition (1) for optimality of permutation π_i for the problem $1||\sum w_iC_i$. Theorem 1 is proven. ■

Uncertain Sequencing Problem

Search of the minimal dominant set *S(T)* for an uncertain problem $1|a_i \le p_i \le b_i|\sum w_iC_i$ may be based on constructing a dominance relation on the set of jobs *J*. To this end, we define a dominance relation as follows.

Definition 2: Job J_u dominates job J_v with respect to T (i.e., $J_u \to J_v$), if there exists a minimal dominant set S(T) for the problem $1|a_i \le p_i \le b_i|\sum w_iC_i$ that each permutation from set S(T) has either the form (..., J_u, J_v, ...) or the form (..., J_u, ..., J_v, ...) (i.e., in a permutation $\pi_k \in S(T)$, job J_u precedes job J_v).

From Definition 2 it follows that minimal dominant set *S(T)* for the deterministic problem $1||\sum w_iC_i$ is a singleton: $\{\pi_k\} = S(T)$. As a result the following dominance relations hold: $J_{k_1} \to J_{k_2} \to ... \to J_{k_n}$. For a general case of the problem $1|a_i \le p_i \le b_i|\sum w_iC_i$, the following claim may be proven using Theorem 1.

Theorem 2: *For the problem* $1|a_i \le p_i \le b_i|\sum w_iC_i$, *job* J_u *dominates job* J_v *with respect to* T *if and only if the following inequality holds:*

$$\frac{w_u}{b_u} \ge \frac{w_v}{a_v}. \tag{3}$$

Due to Theorem 2, if job J_u dominates job J_v and job J_v dominates job J_i, then job J_u dominates job J_i as well. Thus, dominance relation $J_u \to J_v$ is transitive. Theorem 2 allows us to find a minimal dominant set $S(T)$ for the uncertain problem $1|a_i \le p_i \le b_i|\sum w_iC_i$ and to present set *S(T)* in compact form. Indeed via checking condition (3) for each pair of jobs J_u and J_v from the set *J*, we construct digraph $G = (J, A)$ of dominance relation on the set *J*: *Arc* (J_u, J_v) *belongs to set A if and only if dominance relation* $J_u \to J_v$ *holds.* Obviously, it takes $O(n^2)$ time to construct digraph $G = (J, A)$. If due to Theorem 2, linearly ordered set of jobs *J*, $J_{k_1} \to J_{k_2} \to ... \to J_{k_n}$, will be constructed, then set $S(T)$ for the problem $1|a_i \le p_i \le b_i|\sum w_iC_i$ will be a singleton: $\{\pi_k\} = S(T)$. And permutation $\pi_k \in S$ will be optimal for any possible scenario $p \in T$. It is easy to convince that in the case of $\{\pi_k\} = S(T)$, inequality $|A| \ge \frac{n(n-1)}{2}$ must hold for the digraph $G = (J, A)$.

Illustrative Example

Let input data for the instance of the problem $1|a_i \le p_i \le b_i|\sum w_iC_i$ be given in columns 1–4 of Table 1.

Table 1. Input data for the problem $1|a_i \le p_i \le b_i|\sum w_iC_i$

i	a_i	b_i	w_i	w_i / a_i	w_i / b_i
1	1	3	18	18	6
2	5	6	30	6	5
3	4	10	20	5	2
4	3	4	12	4	3
5	4	4	8	2	2
6	5	10	10	2	1
7	2	2	3	1.5	1.5
8	7	10	14	2	1.4

Via testing condition (3) of Theorem 2 for each pair of jobs $J_u \in J$ and $J_v \in J$ we obtain the following relations:

$$\frac{w_1}{b_1} = 6 \geq 6 = \frac{w_2}{a_2};\ \frac{w_2}{b_2} = 5 \geq 5 = \frac{w_3}{a_3};\ \frac{w_2}{b_2} = 5 \geq 4 = \frac{w_4}{a_4};\ \frac{w_3}{b_3} = 2 \geq 2 = \frac{w_5}{a_5};\ \frac{w_4}{b_4} = 3 \geq 2 = \frac{w_5}{a_5};$$

$$\frac{w_5}{b_5} = 2 \geq 2 = \frac{w_6}{a_6};\ \frac{w_5}{b_5} = 2 \geq 1.5 = \frac{w_7}{a_7};\ \frac{w_5}{b_5} = 2 \geq 2 = \frac{w_8}{a_8}.$$

Thus, condition (3) holds for the following ordered pair of jobs: J_1 and J_2; J_2 and J_3; J_2 and J_4; J_3 and J_5; J_4 and J_5; J_5 and J_6; J_5 and J_7; J_5 and J_8. Due to Theorem 2, the following dominance relations hold: job J_1 dominates job J_2; job J_2 dominates jobs J_3 and J_4; job J_3 dominates job J_5; job J_4 dominates job J_5; job J_5 dominates jobs J_6, J_7, and J_8. It is easy to verify that there are no other dominance relations except those that are transitive to the above ones. Therefore, minimal dominant set $S(T)$ for this instance of the problem $1 \mid a_i \leq p_i \leq b_i \mid \sum w_i C_i$ consists of 2!·3! = 12 permutations.

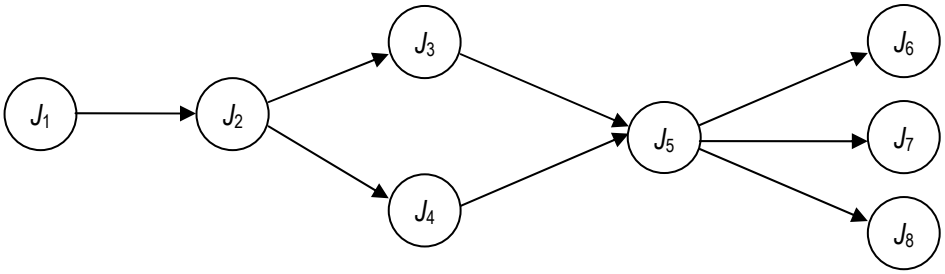


Fig. 1. Digraph $G = (J, A)$ without transitive arcs

Digraph $G = (J, A)$ defining set $S(T) = \{\pi_1, \pi_2, ..., \pi_{12}\}$ is represented in Fig. 1 (for simplicity, the transitive arcs are omitted). Thus, while searching optimal permutation for this instance of the uncertain problem $1 \mid a_i \leq p_i \leq b_i \mid \sum w_i C_i$, it is sufficient to test only 12 permutations (instead of 8! = 256 feasible ones).

Computational Results

In this section, we describe the testing of randomly generated problems $1 \mid a_i \leq p_i \leq b_i \mid \sum w_i C_i$ and answer (by experiments on PC) the question of how many pairs of jobs from set J satisfy condition (3) and how large errors of the optimal values of the criterion $\sum w_i C_i$ are for the schedules constructed using digraph $G = (J, A)$.

The computational algorithm was coded in C++. If relation $J_u \rightarrow J_v$ was fulfilled provided that $u < v$, our algorithm did not tested the validity of the opposite relation: $J_v \rightarrow J_u$. Therefore, an optimal permutation was obtained without fail, if equality $|A| = \frac{n(n-1)}{2}$ was fulfilled for the constructed digraph $G = (J, A)$.

For the experiments, we used an AMD 3000 MHz processor with 1024 MB main memory. We tested random instances of the uncertain problem $1 \mid a_i \leq p_i \leq b_i \mid \sum w_i C_i$ with the following numbers of jobs: $n \in$ {10, 25, 50, 100, 150, ..., 700}. The given integer lower and upper bounds of the possible integer processing times were uniformly distributed in the range [1, 100]. We tested the following errors $L\%$ of the uncertain job processing times: $L \in$ {0.1, 0.5, 1.0, 5.0, 10.0, 15.0, 20.0}. For each job $J_i \in J$, the given lower bound of a job processing time was randomly generated in the range [1, 100] and the upper bound was computed as follows: $b_i = a_i(1 + L\%/100\%)$. For each job $J_i \in J$, the weight $w_i > 0$ was randomly generated in the range [1, 50].

Table 2 represents the computational results for 80 series of the randomly generated instances. Each series included 10 instances with the same combination of the above n and L. The left-hand side of Table 2 (columns 1

– 6) represents the computational results for instances with the numbers of jobs from set {10, 25, 50, 100, 150,…, 300}. The right-hand side of Table 2 (columns 7 – 12) represents the computational results for instances with the number of jobs from set {350, 400,…, 700}. The number of series is given in column 1 (for series numbered from 1 to 40), and in column 7 (for series numbered from 41 to 80). The number of jobs in one instance is given in the corresponding column 2 or 8. The error L of the uncertain job processing times (in percentage) is given in the corresponding columns 3 or 9. The average error of the objective function value $\Phi^0 = \sum_{i=1}^{n} w_i C_i^0$ calculated for the heuristic schedules constructed due to Theorem 2 and digraph $G = (J, A)$, with respect to optimal objective function value $\Phi^* = \sum_{i=1}^{n} w_i C_i^*$, is given in the corresponding columns 4 or 10 (namely, values $(\Phi^0 - \Phi^*) : \Phi^*$ are given in columns 4 and 10). The average relative number of arcs $|A|$ (in percentage) constructed due to validity of condition (3), with respect to the number of arcs in the complete circuit-free digraph, is given in the corresponding columns 5 or 11 (namely, values $(|A| : \frac{n(n-1)}{2})100\%$ are given in columns 5 and 11). The average CPU-time (in seconds) used by the processor AMD 3000 MHz for solving one instance (approximately or exactly) is given in the corresponding columns 6 or 12.

From the experiments, it follows that dominance relation stated in Theorem 2 allow us to solve exactly all the instances from the series with numbers 1 – 10 and 14 (see column 5). The lowest relative number of arcs, i.e. 78.81069%, was constructed for the series with number 75. The lowest average quality of the schedules, i.e. $(\Phi^0 - \Phi^*) : \Phi^* = 0.759582$, was obtained for the series with the largest number 80. The largest CPU-time, 37.1 s, was obtained for the series with number 48. The average quality of the schedules obtained depends of the error L of the job processing times and remains almost the same for the instances with different number of jobs provided that they have the same error $L\%$ of the uncertain processing times. Increasing simultaneously both numbers *n* and L decreases the number of instances solved exactly due to Theorem 2.

Table 2. Computational results for randomly generated instances $1 \mid a_i \le p_i \le b_i \mid \sum w_i C_i$

	n	Error *L* %	Objective error	Number of arcs,%	CPU-time,s		*n*	Error *L* %	Objective error	Number of arcs,%	CPU-time,s
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	10	0.1	0.000000	100.00000	0	41	350	1.0	0.004562	99.181989	1.9
2	25	0.1	0.000000	100.00000	0	42	400	1.0	0.004622	99.160902	3.3
3	50	0.1	0.000000	100.00000	0	43	450	1.0	0.004386	99.175452	5.3
4	100	0.1	0.000000	100.00000	0	44	500	1.0	0.004430	99.179559	8.2
5	150	0.1	0.000000	100.00000	0.1	45	550	1.0	0.004539	99.179467	11.9
6	200	0.1	0.000000	100.00000	0.2	46	600	1.0	0.004494	99.202282	17.1
7	250	0.1	0.000000	100.00000	0.5	47	650	1.0	0.004504	99.197867	24.8
8	300	0.1	0.000000	100.00000	1.1	48	700	1.0	0.004738	99.174290	37.1
9	10	0.5	0.000000	100.00000	0	49	350	5.0	0.051612	94.942939	1.7
10	25	0.5	0.000000	100.00000	0	50	400	5.0	0.052478	94.784712	3.0
11	50	0.5	0.000058	99.991837	0	51	450	5.0	0.052236	94.871171	4.8
12	100	0.5	0.000000	99.997980	0	52	500	5.0	0.052704	94.917756	7.3
13	150	0.5	0.000000	99.996421	0	53	550	5.0	0.051536	95.060772	10.9
14	200	0.5	0.000000	100.00000	0.3	54	600	5.0	0.054459	94.834780	15.2
15	250	0.5	0.000006	99.998394	0.7	55	650	5.0	0.053217	94.908475	21.2
16	300	0.5	0.000033	99.996210	1.7	56	700	5.0	0.050923	94.883303	29.3

17	10	1.0	0.000374	98.888889	0	57	350	10.0	0.196527	89.868522	1.5
18	25	1.0	0.004613	99.333333	0	58	400	10.0	0.189964	89.885965	2.5
19	50	1.0	0.003874	99.412245	0	59	450	10.0	0.199014	89.876169	4.2
20	100	1.0	0.003450	99.236364	0	60	500	10.0	0.198004	89.393587	6.3
21	150	1.0	0.005091	99.314541	0.1	61	550	10.0	0.182142	89.829641	9.4
22	200	1.0	0.004504	99.190452	0.2	62	600	10.0	0.192427	89.771786	13.3
23	250	1.0	0.004385	99.178474	0.5	63	650	10.0	0.191777	89.593694	18.9
24	300	1.0	0.004397	99.196433	0.1	64	700	10.0	0.197116	89.578500	26.9
25	10	5.0	0.010130	92.666667	0	65	350	15.0	0.416224	84.606631	1.3
26	25	5.0	0.044132	95.033333	0	66	400	15.0	0.421506	84.389724	2.2
27	50	5.0	0.033901	95.346939	0	67	450	15.0	0.419705	84.949567	3.6
28	100	5.0	0.049146	95.032323	0	68	500	15.0	0.450631	84.497074	5.5
29	150	5.0	0.050174	94.741834	0	69	550	15.0	0.432102	84.505647	8.1
30	200	5.0	0.050892	95.029648	0.2	70	600	15.0	0.447372	84.360045	11.4
31	250	5.0	0.054236	94.993092	0.5	71	650	15.0	0.418753	84.208131	15.6
32	300	5.0	0.049330	94.978595	0.9	72	700	15.0	0.454469	84.130922	21.6
33	10	10.0	0.129524	89.111111	0	73	350	20.0	0.747840	79.328530	1.2
34	25	10.0	0.124472	89.400000	0	74	400	20.0	0.779839	79.001754	1.8
35	50	10.0	0.179310	90.889796	0	75	450	20.0	0.775389	78.810690	3.0
36	100	10.0	0.201273	89.715152	0	76	500	20.0	0.749320	79.445210	4.7
37	150	10.0	0.199625	89.681432	0.1	77	550	20.0	0.767790	79.044080	6.8
38	200	10.0	0.185334	89.533668	0.1	78	600	20.0	0.755315	79.437340	9.8
39	250	10.0	0.197567	89.108112	0.4	79	650	20.0	0.779930	79.073272	13.4
40	300	10.0	0.190829	89.575920	0.8	80	700	20.0	0.759582	79.465195	18.4

Conclusion

The main issue of this paper is to show how to construct a minimal dominant set $S(T)$ in polynomial time via constructing digraph $G = (J, A)$ as a compact presentation of set $S(T)$ of dominant permutations. We estimated a strength of using minimal dominant set $S(T)$ by extensive computational experiments for randomly generated problems $1 \mid a_i \le p_i \le b_i \mid \sum w_i C_i$ with number n of jobs from the range [10, 700].

Bibliography

[Allahverdi, Sotskov, 2003] A. Allahverdi, Yu.N. Sotskov. Two-machine flowshop minimum-length scheduling with random and bounded processing times, *International Transactions in Operational Research* 10 (2003) 65-76.

[Allahverdi, Aldowaisan, Sotskov, 2003] A. Allahverdi, T. Aldowaisan, Yu.N. Sotskov. Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times, *International Transactions of Mathematical Sciences* 39 (2003) 2475-2486.

[Averbakh, 2000] I. Averbakh. Minmax regret solutions for minmax optimization problems with uncertainty, *Operations Research Letters* 27 (2000) 57-65.

[Averbakh, 2001] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty, *Mathematical Programming,* Series A 90 (2001) 263-272.

[Brasel, Sotskov, Werner, 1996] H.-M. Brasel, Yu. N. Sotskov, F. Werner. Stability of a schedule minimizing mean flow time, *Mathematical and Computer Modelling* 24 (10) (1997) 39-53.

[Daniels, Kouvelis, 1995] R.L. Daniels, P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single-stage production, *Management Science* 41 (2) (1995) 363-376.

[Graham et al., 1976] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics* 5 (1976) 287-326.

[Garey, Johnson, 1979] M.R. Garey, D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* San Francisko: Freeman, USA, 1979.

[Kouvelis, Yu, 1997] P. Kouvelis, G. Yu. *Robust Discrete Optimization and Its Applications*. Boston: Kluwer Academic Publishers, USA, 1997.

[Lebedev, Averbakh, 2006] V. Lebedev, I. Averbakh. Complexity of minimizing the total flow time with interval data and minmax regret criterion, *Discrete Applied Mathematics* 154 (2006) 2167-2177.

[Leshchenko, Sotskov, 2007] N. Leshchenko, Yu. N. Sotskov. Realization of an optimal schedule for the two-machine flow-shop with interval job processing times, *International Journal "Information, Theory & Applications"* 14 (2007) 182-189.

[Lai, Sotskov, 1999] T.-C. Lai, Yu. N. Sotskov. Sequencing with uncertain numerical data for makespan minimization, *Journal of the Operational Research Society* 50 (1999) 230-243.

[Pinedo, 1995] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Prentice-Hall: Enlewood Cliffs, USA, 1995.

[Smith, 1956] W.E. Smith, Various optimizers for single-stage production, *Naval Research and Logistics Quarterly* 3 (1) (1956) 59-66.

[Sotskov, Allahverdi, Lai, 2004] Yu.N. Sotskov, A. Allahverdi, T.-C. Lai. Flowshop scheduling problem to minimize total completion time with random and bounded processing times, *Journal of the Operational Research Society* 55 (2004) 277-286.

[Sotskov, Sotskova, 2004] Yu. N. Sotskov, N. Sotskova. *Scheduling Theory: Systems with Uncertain Numerical Parameters,* Minsk: United Institute of Informatics Problems, Belarus, 2004 (in Russian).

[Sotskov, Sotskova, Werner, 1997] Yu. N. Sotskov, N. Sotskova, F. Werner. Stability of an optimal schedule in a job shop, *Omega - Journal of Operational Research* 25 (4) (1997) 245-280.

[Sotskov, Wagelmans, Werner, 1998] Yu.N. Sotskov, A.P.M. Wagelmans, F. Werner. On the calculation of the stability radius of an optimal or an approximate schedule, *Annals of Operations Research* 83 (1998) 213-252.

[Tanaev, Sotskov, Strusevich, 1994] V.S. Tanaev, Yu.N. Sotskov, V.A. Strusevich. *Scheduling Theory: Multi-Stage Systems,* Dordrecht: Kluwer Academic Publishers, Netherlands, 1994.

[Yang, Yu, 2002] J. Yang, G. Yu. On the robust single machine scheduling problem, *Journal of Combinatorial Optimization* 6 (2002) 17-33.

Authors' Information

Yuri N. Sotskov – *Professor, DSc, PhD, United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: sotskov@newman.bas-net.by*

Natalja G. Egorova – *Junior researcher, United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Surganova str., 6, 220012, Minsk, Belarus; e-mail: egorova@newman.bas-net.by*

MULTIDIMENSIONAL HETEROGENEOUS VARIABLE PREDICTION BASED ON EXPERTS' STATEMENTS*

Gennadiy Lbov, Maxim Gerasimov

Abstract: *In the works [1, 2] we proposed an approach of forming a consensus of experts' statements for the case of forecasting of qualitative and quantitative variable. In this paper, we present a method of aggregating sets of individual statements into a collective one for the general case of forecasting of multidimensional heterogeneous variable.*

Keywords: *multidimensional variable, expert statements, coordination.*

ACM Classification Keywords: *I.2.6. Artificial Intelligence - knowledge acquisition.*

Conference: *The paper is selected from XIV[th] International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Introduction

Let Γ be a population of elements or objects under investigation. By assumption, L experts give predictions of values of unknown *m*-dimensional heterogeneous feature Y for objects $a \in \Gamma$, being already aware of their description $X(a)$. We assume that $X(a) = (X_1(a),...,X_j(a),...,X_n(a))$, $Y(a) = (Y_1(a),...,Y_j(a),$ $...,Y_m(a))$, where the sets X and Y may simultaneously contain qualitative and quantitative features X_j, $j = \overline{1,n}$; or Y_j, $j = \overline{1,m}$; respectively. Let D_j^X be the domain of the feature X_j, $j = \overline{1,n}$, D_j^Y be the domain of the feature Y_j, $j = \overline{1,m}$. The feature spaces are given by the product sets: $D^X = \prod_{j=1}^n D_j^X$ and $D^Y = \prod_{j=1}^m D_j^Y$. By assumption, exactly combination of values $Y_1(a),\ldots,Y_j(a),\ldots,Y_m(a)$ is important, so we have to estimate the whole set Y simultaneously.

We shall say that a set E is a *rectangular set* in D^X if $E = \prod_{j=1}^n E_j$, $E_j \subseteq D_j^X$, $E_j = [\alpha_j, \beta_j]$ if X_j is a quantitative feature, E_j is a finite subset of feature values if X_j is a nominal feature. In the same way rectangular sets in D^Y are defined.

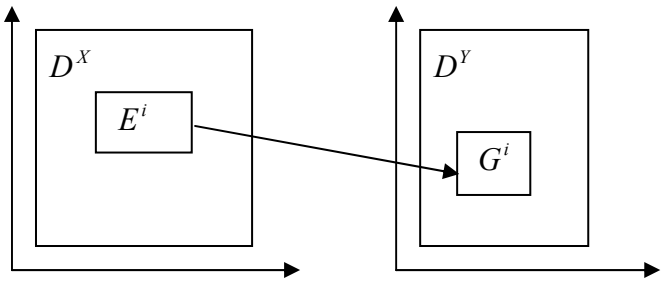


Fig. 1.

* The work was supported by the RFBR under Grant N07-01-00331a.

In this paper, we consider statements S^i, $i=\overline{1,M}$; represented as sentences of type "if $X(a)\in E^i$, then $Y(a)\in G^i$", where E^i is a rectangular set in D^X, G^i is a rectangular set in D^Y (see Fig. 1). By assumption, each statement S^i has its own weight w^i ($0<w^i\le 1$ for individual statements). Such a value is like a measure of "confidence".

Let us remark that the statement "if $X(a)\in E$, then $Y(a)\in D^Y$" is equal to the statement "I know nothing about $Y(a)$ if $X(a)\in E$".

Without loss of generality we may assume that experts themselves have equal "weights".

Setting of a Problem

We begin with some definitions.

Denote by $E^{i_1 i_2} := E^{i_1}\oplus E^{i_2}=\prod_{j=1}^{n}(E_j^{i_1}\oplus E_j^{i_2})$, where $E_j^{i_1}\oplus E_j^{i_2}$ is the *Cartesian join* of feature values $E_j^{i_1}$ and $E_j^{i_2}$ for feature X_j and is defined as follows. When X_j is a nominal feature, $E_j^{i_1}\oplus E_j^{i_2}$ is the union: $E_j^{i_1}\oplus E_j^{i_2}=E_j^{i_1}\bigcup E_j^{i_2}$. When X_j is a quantitative feature, $E_j^{i_1}\oplus E_j^{i_2}$ is a minimal closed interval such that $E_j^{i_1}\bigcup E_j^{i_2}\subseteq E_j^{i_1}\oplus E_j^{i_2}$ (see Fig. 2).

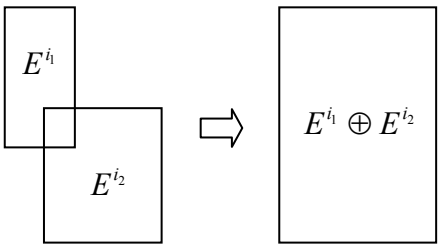


Fig. 2.

In the work [3] we proposed a method to measure the distances between sets (e.g., E^1 and E^2) in heterogeneous feature space. Consider some modification of this method. By definition, put

$\rho(E^1,E^2)=\sum_{j=1}^{n}k_j\rho_j(E_j^1,E_j^2)$ or $\rho(E^1,E^2)=\sqrt{\sum_{j=1}^{n}k_j(\rho_j(E_j^1,E_j^2))^2}$, where $0\le k_j\le 1$, $\sum_{j=1}^{n}k_j=1$.

Values $\rho_j(E_j^1,E_j^2)$ are given by: $\rho_j(E_j^1,E_j^2)=\dfrac{|E_j^1\Delta E_j^2|}{|D_j^X|}$ if X_j is a nominal feature, $\rho_j(E_j^1,E_j^2)=\dfrac{r_j^{12}+\theta|E_j^1\Delta E_j^2|}{|D_j^X|}$ if X_j is a quantitative feature, where $r_j^{12}=\left|\dfrac{\alpha_j^1+\beta_j^1}{2}-\dfrac{\alpha_j^2+\beta_j^2}{2}\right|$. It can be proved that the triangle inequality is fulfilled if and only if $0\le\theta\le 1/2$.

The proposed measure ρ satisfies the requirements of distance there may be. Note that we can use another measure of differences (for example, see [4]).

In this paper we assume that distance between rectangular sets in D^Y is known.

Consider some "natural" algorithm of forming a consensus of experts' statements (denote it by A).

Let for some point $x \in D^X$ we have two statements S^1 and S^2 with the weights w^1 and w^2. Suppose G^1 and G^2 are the images prescribed by these statements to the point x.

If $\rho(G^1, G^2) < \varepsilon$, where ε is a threshold, then it may be assumed that the set $G^1 \oplus G^2$ is "naturally" prescribed to the point x. Note that if these statements are given by different experts, then we more confidence in resulted statement, so the weight of this statement is higher than w^1 and w^2 (it may be even more than 1).

Otherwise, if $\rho(G^1, G^2) \geq \varepsilon$, then it may be assumed that only one statement with higher weight is remained and our confidence in it (and the weight of it) is decreased.

If for some point $x \in D^X$ we have more than two statements, the algorithm A coordinates them in the same way.

Since there are M statements, we have up to 2^M sets in D^X with different prescribed images. These sets are in the form of E_1 or $E_1 \setminus (E_2 \bigcup E_3 \ldots)$, where E_i are rectangular sets in D^X.

Consider algorithms B of forming a consensus of experts' statements under restrictions on amount of resulted statements. The value $F(B) = \int_{D^X} (\rho(G_A(x), G_B(x))^2 dx$ estimates a quality of the algorithm B. Here $G_A(x)$, $G_B(x)$ are the images prescribed to the point $x \in D^X$ by algorithms A and B, respectively. In the general case, the best algorithm $B^* = \arg\min_B F(B)$ is unknown. Further on, the heuristic algorithm of forming a consensus of experts' statements is considered.

Preliminary Analysis

We first treat each expert's statements separately for rough analysis. Let us consider some special cases.

Case 1 ("coincidence"): $\max_j \max(\rho_j(E^{i_1}, E^{i_1} \oplus E^{i_2}), \rho_j(E^{i_2}, E^{i_1} \oplus E^{i_2})) < \delta$ and $\rho(G^{i_1}, G^{i_2}) < \varepsilon_1$, where δ, ε_1 are thresholds decided by the user, $i_1, i_2 \in \{1,\ldots,M\}$. In this case we unite statements S^{i_1} and S^{i_2} into resulting one: "if $X(a) \in E^{i_1} \oplus E^{i_2}$, then $Y(a) \in G^{i_1} \oplus G^{i_2}$".

Case 2 ("inclusion"): $\min(\max_j(\rho_j(E^{i_1}, E^{i_1} \oplus E^{i_2})), \max_j(\rho_j(E^{i_2}, E^{i_1} \oplus E^{i_2}))) < \delta$ and $\rho(G^{i_1}, G^{i_2}) < \varepsilon_1$, where $i_1, i_2 \in \{1,\ldots,M\}$. In this case we unite statements S^{i_1} and S^{i_2} too: "if $X(a) \in E^{i_1} \oplus E^{i_2}$, then $Y(a) \in G^{i_1} \oplus G^{i_2}$".

Case 3 ("contradiction"): $\max_j \max(\rho_j(E^{i_1}, E^{i_1} \oplus E^{i_2}), \rho_j(E^{i_2}, E^{i_1} \oplus E^{i_2})) < \delta$ and $\rho(G^{i_1}, G^{i_2}) > \varepsilon_2$, where ε_2 is a threshold decided by the user, $i_1, i_2 \in \{1,\ldots,M\}$. In this case we exclude both statements S^{i_1} and S^{i_2} from the list of statements.

Coordination of Similar Statements

Consider the list of l-th expert's statements after preliminary analysis $\Omega_1(l) = \{S^1(l),\ldots, S^{m_l}(l)\}$. Denote by $\Omega_1 = \bigcup_{l=1}^{L} \Omega_1(l)$, $M_1 = |\Omega_1|$.

Determine now distance between rectangular sets in D^X. Determine values k_j from this reason: if far sets G^{i_1} and G^{i_2} corresponds to far sets $E_j^{i_1}$ and $E_j^{i_2}$, then the feature X_j is more "valuable" than another features,

hence, value k_j is higher. We can use, for example, these values: $k_j = \frac{\tau_j}{\sum_{i=1}^{n} \tau_i}$, where $\tau_j = \sum_{u=1}^{M_1} \sum_{v=1}^{M_1} \rho(G^u, G^v) \rho_j(E_j^u, E_j^v)$, $j = \overline{1,n}$.

Denote by $r^{i_1 i_2} := d(E^{i_1 i_2}, E^{i_1} \bigcup E^{i_2})$.

The value $d(E,F)$ is defined as follows: $d(E,F) = \max\limits_{E' \subseteq E \setminus F} \min\limits_{j} \frac{k_j \mid E'_j \mid}{diam(E)}$, where E' is any rectangular set (see Fig. 3), $diam(E) = \max\limits_{x,y \in E} \rho(x,y)$.

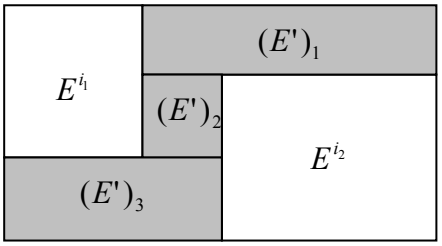


Fig. 3.

By definition, put $I_1 = \{\{1\},...,\{M_1\}\},...,$ $I_q = \{\{i_1,...,i_q\} \mid r^{i_u i_v} \le \delta$ and $\rho(G^{i_u}, G^{i_v}) < \varepsilon_1 \quad \forall u, v = \overline{1,q}\}$, where δ, ε_1 are thresholds decided by the user, $q = \overline{2,Q}$; $Q \le M_1$. Let us remark that the requirement $r^{i_u i_v} \le \delta$ is like a criterion of "insignificance" of the set $E^{uv} \setminus (E^{i_u} \bigcup E^{i_v})$. Notice that someone can use another value d to determine value r, for example:

$$d(E,F,G) = \max_{E' \subseteq E \setminus (F \cup G)} \frac{\min(diam(F \oplus E') - diam(F), diam(G \oplus E') - diam(G))}{diam(E)}.$$

Further, take any set $J_q = \{i_1,...,i_q\}$ of indices such that $J_q \in I_q$ and $\forall \Delta = \overline{1, Q-q} \quad \forall J_{q+\Delta} \in I_{q+\Delta}$ $J_q \not\subset J_{q+\Delta}$. Now, we can aggregate the statements S^{i_1}, ..., S^{i_q} into the statement S^{J_q}:

S^{J_q} ="if $X(a) \in E^{J_q}$, then $Y(a) \in G^{J_q}$", where $E^{J_q} = E^{i_1} \oplus ... \oplus E^{i_q}$, $G^{J_q} = G^{i_1} \oplus ... \oplus G^{i_q}$.

By definition, put to the statement S^{J_q} the weight $w^{J_q} = \frac{\sum_{i \in J_q} c^{iJ_q} w^i}{\sum_{i \in J_q} c^{iJ_q}}$, where $c^{iJ_q} = 1 - \rho(E^i, E^{J_q})$.

The procedure of forming a consensus of single expert's statements consists in aggregating into statements S^{J_q} for all J_q under previous conditions, $q = \overline{1,Q}$.

Let us remark that if, for example, $k_1 < k_2$, then the sets E_1 and E_2 (see Fig. 4) are more suitable to be united (to be precise, the relative statements), than the sets F_1 and F_2 under the same another conditions.

Note that we can consider another criterion of unification (instead of $r^{i_u i_v} \le \delta$): aggregate statements S^{i_1}, ..., S^{i_q} into the statement S^{J_q} only if $w^{J_q} > \varepsilon'$, where ε' is a threshold decided by the user.

After coordinating each expert's statements separately, we can construct an agreement of several independent experts. The procedure is as above, except the weights: $w^{J_q} = \sum_{i \in J_q} c^{iJ_q} w^i$ (the more experts give similar statements, the more we trust in resulted statement).

Denote the list of statements after coordination by Ω_2, $M_2 := |\Omega_2|$.

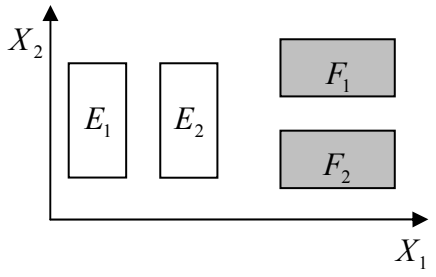


Fig. 4.

Coordination of Non-similar Statements

After constructing of a consensus of similar statements, we must form decision rule in the case of intersected non-similar statements. The procedure in such cases is as follows.

To each $h = \overline{2, M_2}$ consider statements $S^{(1)},...,S^{(h)} \in \Omega_2$ such that $\widetilde{E}^h := E^{(1)} \bigcap ... \bigcap E^{(h)} \neq \varnothing$, where $E^{(i)}$ are related sets to statements $S^{(i)}$.

Denote $I(l) = \left\{ i \middle| S^i(l) \in \Omega_1(l), \quad E^i(l) \bigcap \widetilde{E}^h \neq \varnothing \right\}$, where $E^i(l)$ are related sets to statements $S^i(l)$.

Consider related sets $G^i(l)$, where $l = \overline{1, L}$; $i \in I(l)$. Denote by $w^i(l)$ the weights of statements $S^i(l)$.

As above, unite sets $G^{(i_1)}(l_1),...,G^{(i_q)}(l_q)$ if $\rho(G^{i_u}, G^{i_v}) < \varepsilon_1 \ \ \forall u, v = \overline{1, q}$. Denote by $\widetilde{G}^1,...,\widetilde{G}^\lambda,...,\widetilde{G}^\Lambda$ the sets after procedure of unification of the sets $G^i(l)$. Consider the statements $\widetilde{S}^\lambda$: "if $X(a) \in \widetilde{E}^h$, then $Y(a) \in \widetilde{G}^\lambda$".

In order to choose the best statement, we take into consideration these reasons:

1) similarities between sets $\widetilde{E}^h$ and $E^i(l)$;
2) similarities between sets $\widetilde{G}^\lambda$ and $G^i(l)$;
3) weights of statements $S^i(l)$;
4) we must distinguish cases when similar / contradictory statements produced by one or several experts.

We can use, for example, such values: $w^\lambda = \sum_{l=1}^{L} \frac{\sum_{i \in I(l)} (1 - \rho(G^{(i)}(l), \widetilde{G}^{(\lambda)}))(1 - \rho(E^{(i)}(l), \widetilde{E}^h)^2 w^i(l)}{\sum_{i \in I(l)} (1 - \rho(E^{(i)}(l), \widetilde{E}^h)}$.

Denote by $\lambda^* := \arg\max_\lambda w^\lambda$.

Thus, we can make decision statement: $\widetilde{S}^h =$"if $X(a) \in \widetilde{E}^h$, then $Y(a) \in \widetilde{G}^{\lambda^*}$" with the weight $\widetilde{w}^h := w^{\lambda^*} - \max_{\lambda \neq \lambda^*} w^\lambda$.

Denote the list of such statements by Ω_3.

Final decision rule is formed from statements in Ω_2 and Ω_3.

Conclusion

Suggested method of forming of united decision rule can be used for coordination of several experts statements, and different decision rules obtained from learning samples and/or time series. Notice that we can range resulted statements by their weights, and then exclude "ignorable" statements from decision rule or inquire for more information for corresponding sets from experts.

Bibliography

[1] G.Lbov, M.Gerasimov. Constructing of a Consensus of Several Experts Statements. In: Proc. of XII Int. Conf. "Knowledge-Dialogue-Solution", 2006, pp. 193-195.

[2] G.Lbov, M.Gerasimov. Interval Prediction Based on Experts' Statements. In: Proc. of XIII Int. Conf. "Knowledge-Dialogue-Solution", 2007, Vol. 2, pp. 474-478.

[3] G.S.Lbov, M.K.Gerasimov. Determining of Distance Between Logical Statements in Forecasting Problems. In: Artificial Intelligence, 2'2004 [in Russian]. Institute of Artificial Intelligence, Ukraine.

[4] A.Vikent'ev. Measure of Refutation and Metrics on Statements of Experts (Logical Formulas) in the Models for Some Theory. In: Int. Journal "Information Theories & Applications", 2007, Vol. 14, No.1, pp. 92-95.

Authors' Information

Gennadiy Lbov - *Institute of Mathematics, SB RAS, Koptyug St., bl.4, Novosibirsk, Russia;*
e-mail: lbov@math.nsc.ru

Maxim Gerasimov - *Institute of Mathematics, SB RAS, Koptyug St., bl.4, Novosibirsk, Russia,*
e-mail: max_post@ngs.ru

A METAONTOLOGY FOR MEDICAL DIAGNOSTICS OF ACUTE DISEASES. PART 1. AN INFORMAL DESCRIPTION AND DEFINITIONS OF BASIC TERMS

Mary Chernyakhovskaya, Alexander Kleshchev, Phillip Moskalenko

***Abstract**: The aim of this article is to describe formally a metaontology for medical diagnostics of acute diseases in the language of applied logic. The article includes an informal description of the metaontology, and the part of its model which contains the definitions for basic terms of knowledge and situations and also their integrity restrictions in the form of ontological agreements.*

***Keywords**: Medical Diagnostics, ontology model, metaontology.*

***ACM Classification Keywords**: I.2.1 Applications and Expert Systems, I.2.4 Knowledge Representation Formalisms and Methods, J.3 Life and Medical Sciences.*

***Conference**: The paper is selected from International Conference "Intelligent Information and Engineering Systems" INFOS 2008, Varna, Bulgaria, June-July 2008*

Introduction

Any expert system is based on a conceptualization of the domain. An explicit representation of the conceptualization is usually called ontology [1]. Publications of domain ontologies and their models representing the conceptualizations which are near to the ones used in science, education and practical activities are of particular interest. There are three reasons for this. First, such publications extend our view of complex ontology construction. Second, they bring out in which direction the languages for representation of complex ontology models should be developed. Third, the published ontologies and their simplifications can be used in expert systems development.

These ideas relate to expert systems for medical diagnostics in full measure. The ontologies which are the basis for some expert systems of medical diagnostics are described in sufficient detail [2], sometimes in a formal manner [3]. But more often it is rather difficult to extract the ontology that is the basis from publications on expert systems [4, 5]. The ontologies which are the basis for many expert systems of medical diagnostics are significantly simplified ones in comparison with the real conceptualizations of this domain. Such properties as development of pathological processes in time, interaction of various types of cause-and-effect relations, and also combined and complicated pathologies are usually not considered in these ontologies.

Studying real ontologies of medical diagnostics (which are different in details for medicine of different countries, as it is fairly noted in [1]) goes back to [6, 7]. At present time various means for ontology description in form of computer languages [8, 9] as well as mathematical ones [10] have been developed. The development of means for ontology description led to the publication of a formal model of medical diagnostics ontology that was close to real conceptions in medicine [11].

Medical diagnostics is a wide domain. It means that ontologies of many its divisions are special cases of the same metaontology of this domain. The aim of this article is an explicit and formal description of this metaontology in the language of applied logic [10].

This paper was made according to the project of RFBR № 06-07 89071 «An investigation of possibilities for collective managing information resources of various levels of generality in the semantic Web » and to the project of FEBRAS № 06-III-A-01-457 «Designing, implementing and developing the bank of medical knowledge in the Internet network».

1. An informal description of the metaontology

In this article a metaontology of medical diagnostics is the object of modeling. The metaontology is a set of conceptual ideas about the processes in a patient's body and in its environment which are essential for solving the task of medical diagnostics. These ideas are based on the results of the works [7, 11].

The reality of the domain is a set of situations [12]. Every situation contains information about patient's body during a time interval. The beginning of this interval is the earliest moment to which the information about this situation is related. Time moments are measured in hours from this beginning using the scale of nonnegative integers.

The processes which proceed in a patient's body can be arbitrary divided into external (observable) and internal ones. The latter processes are the object of diagnostics. The observable processes are called signs, and the internal ones are called diseases. The signs have values which can be obtained during their examination and vary in time. They are considered as qualitative (scalar)[1]. The signs are a subclass of the observations class. Another subclass of the observations class is the anatomical-and-physiological features of a patient's body. They also have scalar values. In this article it is assumed that these values cannot vary in time. The last subclass of the observations class is the events which happened to a patient. They also have scalar values. The events can happen at individual time moments and the values of an event can be different at different time moments.

In this article only the acute diseases are considered. A patient can be healthy or have one or more diseases. Every disease proceeds in time and can sequentially pass through several stages in its development. They are called development periods. The diagnosis is a set of the diseases with which a patient is ill in the situation. Every disease from the diagnosis can have a single cause.

The basic type of the association between the processes which proceed in a patient's body is the class of cause-and-effect relations. This class includes complications, etiologies, clinical manifestations, clinical manifestations modified by event's influence, responses to event's influence and normal reactions. A cause-and-effect relation develops in time according to one of the possible variants of its development which is determined by the values of acting factors (anatomical-and-physiological features) and perhaps by the cause.

A complication associates a (primary) disease from the diagnosis (the cause) with another (secondary) disease from the diagnosis (the effect). The secondary disease arises as a complication of the primary disease in a time lapse after its beginning. A variant of a complication depends on the values of the acting factors only.

An etiology associates a value of an event (the cause-event) with a disease from the diagnosis (the effect). The disease arises as an effect of this event in a time lapse after the event happened. A variant of an etiology depends on the value of the cause-effect and on the values of the acting factors.

A clinical manifestation associates a disease from the diagnosis (the cause) with a sign (the effect). A variant of the clinical manifestation development depends on a development period of the disease and on the values of the acting factors. The values of the sign can be the effect of the disease on the time interval which corresponds to a diseases' development period. This development period in its turn can be divided into several dynamics periods which are determined by a variant of the clinical manifestation's development.

A clinical manifestation modified by an event's influence has the cause that is a disease from the diagnosis and the cause-event with a value. Its effect is a sign. A variant of a clinical manifestation's development modified by an event's influence depends on the value of the cause-event and on the values of the acting factors. The values of the sign can be the effect of the combined action of the cause and cause-event on a time interval which begins at the moment when the event happens. This time interval in its turn can be divided into several dynamics periods which are determined by a variant of the clinical manifestation's development modified by the event's influence.

---

[1] In medical diagnostics the values of signs (and also of events and anatomical-and-physiological features) can also be quantitative (dimensional) and be represented by integer or rational numbers. But in this article only scalar values will be considered in order to shorten the description of the metaontology.

A response to an event's influence associates a value of an event (the cause-event) with a sign (the effect). A variant of the development of a response to an event's influence depends on the value of the cause-event and on the values of the acting factors. The values of the sign can be the effect of the cause-event on a time interval which begins at the moment when the event happens. This time interval in its turn can be divided into several dynamics periods which are determined by a variant of the development of the response to the event's influence.

The cause of a normal reaction is not identified and its effect is a sign. A variant of a normal reaction's development depends on the values of the acting factors only. During the time intervals when the values of the sign do not have other causes they are the effect of the normal reaction.

The values of a sign can vary as a result of the simultaneous influence of several cause-and-effect relations. The whole time interval of each sign's examination can be divided into such periods that during each period the values of the sign are the effect of a single cause-and-effect relation (from all the possible ones). As this takes place, the beginning of the time interval during which a cause-and-effect relation acts can be only the beginning of such a period, and the end of the time interval during which the cause-and-effect relation acts can be only the end of such a period. Among different types of cause-and-effect relations a partial order is established which is determined by the modality[1] of cause-and-effect relations and by the moments of the cause's initiation.

The domain knowledge consists of knowledge about all the observations, diseases and cause-and-effect relations. Knowledge about an observation includes the range of its possible values. Knowledge about a disease includes a sequence of its development periods. Every period contains an interval of the durations of this period. In reality the duration of every development period belongs to the interval of possible durations which corresponds to this period.

In addition, knowledge about cause-and-effect relations, signs and diseases can contain necessary conditions. The fulfillment of such a condition is necessary, so that the associated sign, disease or cause-and-effect relation can take place for a patient. A necessary condition is a conjunction of components. Every component is a reference to an anatomical-and-physiological feature, and also to a subset of possible values of this feature. A component of a condition is considered to be fulfilled if the examined value of this feature belongs to the subset of possible values of this feature. If the necessary condition is absent in the description of a concept then it is considered to be always fulfilled.

Knowledge about any cause-and-effect relation includes the causal regularity that is a disjunction of variants. If a cause-and-effect relation takes place in reality then one of these variants is fulfilled. In general case a variant is an implication. Its antecedent can contain a condition on the cause, a condition on the cause-event, and a condition on the acting factors[2], and its consequent can contain either an interval of possible durations for the time lapses between the moment when the cause took place and the beginning of the disease (of the effect) or a sequence of the dynamics periods[3].

The description of a complication includes the references to the cause (a primary disease), the effect (a secondary disease), the acting factors, the modality and the causal regularity. A variant of the causal regularity is an implication. Its antecedent can contain a condition on the acting factors and its consequent contains an interval of possible durations for the time lapses between the beginning of the primary disease and the beginning of the secondary disease. The antecedent of such implication is fulfilled if the condition on the acting factors is

---

[1] The modality can take one of two values: *necessity* or *possibility*. The value of *necessity* means that if there is a cause or cause-event for a patient then the cause-and-effect relation necessarily takes place. The value of *possibility* means that cause (or cause-event) does not need to rise to the cause-and-effect relation.

[2] A condition on the acting factors has the same structure as a necessary condition. A condition on the acting factors is fulfilled if each its component is fulfilled (if the condition is absent then it is considered to be always fulfilled).

[3] A dynamics period contains an interval of possible durations for this dynamics period and a set of values of the sign which are possible in this period.

fulfilled. The consequent of the implication is fulfilled if the difference between the time moments when the primary disease and the secondary disease began belongs to the interval of possible durations of this time lapse.

The description of an etiology includes the references to the cause-event, the effect (a disease), the acting factors, the modality and the causal regularity. Its variant is an implication. Its antecedent contains a condition on the cause-event and can contain a condition on the acting factors. Its consequent contains an interval of possible durations for time lapses between the moment when the cause took place and the beginning of the disease (of the effect). A condition on a cause-event is a subset of the range of the possible values for the event. A condition on a cause-event is fulfilled if the value of the cause-event belongs to this subset. The antecedent of an implication is fulfilled if both the condition on the cause-event and the condition on the acting factors are fulfilled. The consequent of an implication is fulfilled if the difference between the time moments when the cause-event happened and the disease began belongs to the interval of possible durations of this time lapse.

The description of a clinical manifestation includes the references to the cause (a disease), to the effect (a sign) to the acting factors, and also the modality and the causal regularity for every development period of the disease. A variant of the causal regularity is an implication. The antecedent of the implication can contain a condition on the acting factors, and its consequent contains a sequence of dynamics periods. The antecedent of the implication is fulfilled if the condition on the acting factors is fulfilled. The consequent of the implication is fulfilled if there is such a partition of the time interval of the development period for the disease into dynamics periods that the duration of every dynamics period belongs to the interval of possible durations for this period, and all the values of the sign which were examined at the moments from this period belong to the set of values which are possible in this period.

The description of a clinical manifestation modified by an event's influence includes the references to the cause-event, to the cause (a disease), to the effect (a sign), to the acting factors, and also the modality and the causal regularity. Its variant is an implication. The antecedent of the implication contains a condition on the cause-event, and can contain a condition on the acting factors. The consequent of the implication contains a sequence of dynamics periods. The antecedent of the implication is fulfilled if both the condition on the cause-event and the condition on the acting factors are fulfilled. The consequent of the implication is fulfilled if there is such a partition of a time interval, that began at the moment when the event happened and the disease was proceeding, into dynamics periods that the duration of every dynamics period belongs to the interval of possible durations of this period, and the values of the sign which were examined at the time moments from this period belong to the set of values which are possible in this period.

The description of a response to an event's influence includes the references to the cause-event, to the effect (a sign), to the acting factors, and also the modality, and the causal regularity. Its variant is an implication. The antecedent of the implication contains a condition on the cause-event, and can contain a condition on the acting factors. The consequent of the implication contains a sequence of dynamics periods. The antecedent of the implication is fulfilled if both the condition on the cause-event and the condition on the acting factors are fulfilled. The consequent of the implication is fulfilled if there is such a partition of a time interval, that began at the moment when the event happened, into dynamics periods that the duration of every dynamics period belongs to the interval of possible durations for this period, and the values of the sign which were examined at the moments from this period belong to the set of values which are possible in this period.

The description of a normal reaction consists of the references to the effect (a sign), to the acting factors, and also of the causal regularity. Its variant is an implication. Its antecedent can contain a condition on the acting factors, and its consequent contains the set of the normal values of the sign (of the effect). The antecedent of the implication is fulfilled if the condition on the acting factors is fulfilled. The consequent of the implication is fulfilled if all the values of the sign which were examined at the moments from the interval when the normal reaction acted belong to the set of the normal values.

2. An extension of the language of applied logic and an applied logic theory which are used in this article

In the article the model of the metaontology of medical diagnostics is represented in the language of applied logic [10] with the use of extensions which were described in the same article. In addition, another specialized extension of the language that is called *categories* is introduced below. Also at this point a modernized variant of the applied logic theory that is called *a definition of partitions* is presented. The original variant of this theory was given in [10].

2.1. The extension "Categories"

2.1.1. The construction $(s_1 \to t_1, s_2 \to t_2, \ldots, s_m \to t_m)$ is a term of this extension. Here $s_1, s_2, \ldots, s_m$ are names, and $t_1, t_2, \ldots, t_m$ are terms. The values of these terms are sets. The value of this term is the set of structural values that is the domain of all the possible mappings with names $s_1, s_2, \ldots, s_m$. The ranges of these mappings are the values of terms $t_1, t_2, \ldots, t_m$ respectively.

2.1.2. The mappings with names $s_1, s_2, \ldots, s_m$ are called *attributes*, and the values of these mappings for a concrete structural value are called *the values of the attributes* for this structural value.

2.1.3. If x is a structural value that belongs to the value of term $(s_1 \to t_1, \ldots, s_i \to t_i, \ldots, s_m \to t_m)$ then any s_i which is a part of terms $t_1, \ldots, t_m$ is considered as a term. The value of this term is the same as the value of term $s_i(x)$.

2.2. The applied logical theory "Definition of partitions"

The applied logical theory *Definition of partitions*(ST, *Intervals*, *Mathematical quantifiers*) contains only the descriptions of name's values:

2.2.1. *"Partitions"* is the set of all possible partitions for the set of nonnegative integers. Every partition is a finite strictly increasing sequence.

$$\textit{partitions} \equiv (\cup \text{ (length: I[0, } \infty)) \{(\text{sequence: I} \Uparrow \text{ (length+1)) (\&(element : I[1, length])}$$
$$\pi(\text{element, sequence}) < \pi(\text{element+1, sequence}))\})$$

2.2.2. *"Element"* is a function; its arguments are a partition and an integer in the range from 0 to the number of elements in the partition; its result is the element of this partition with the number which is equal to the second argument.

$$\textit{element} \equiv (\lambda(\text{partition: } \textit{partitions}) \text{ (element: I[0, } \textit{length}(\text{partition})-1)]) \ \pi(\text{element+1, partition}))$$

2.2.3. *"Interval"* is a function; its arguments are a partition and a positive integer which is less than the number of elements in this partition; its result is the interval of nonnegative integers between the element of this partition with the number which is equal to the second argument and the element with the previous number.

$$\textit{interval} \equiv (\lambda(\text{partition: } \textit{partitions}) \text{ (element: I[1, } \textit{length}(\text{partition})-1)])$$
$$\text{I[}\textit{element}(\text{partition, element}-1), \textit{element}(\text{partition, element})])$$

3. The basic concepts and ontological agreements which define knowledge and reality of the domain

In this section all the classes of observations and diseases, and also the concepts of knowledge and reality which are related to them are described.

3.1. The basic concepts and ontological agreements which define knowledge of the domain (the parameters of the metaontology model)

In this section the basic terms which are used for description of the domain knowledge, and also the restrictions on their values are introduced. These restrictions do not depend on the values of the terms for defining reality.

3.1.1. "*Signs*" is a class of concepts which correspond to observable processes. The values of signs are determined by one of four possible cause-and-effect relations. Knowledge must contain at least one sign.

$$\text{sort } signs\text{: } \{\}\mathbb{N} \setminus \{\varnothing\}$$

3.1.2. "*Events*" is a class of concepts which correspond to events that can happen to patients, and that should be taken into account during diagnosing.

$$\text{sort } events\text{: } \{\}\mathbb{N}$$

3.1.3. "*Features*" is a class of concepts which correspond to anatomical-and-physiological features of patients that should be taken into account during diagnosing.

$$\text{sort } features\text{: } \{\}\mathbb{N}$$

3.1.4. The names of all the signs, events and features are different.

$$\text{signs} \cap \text{events} = \varnothing \;\&\; \text{features} \cap \text{events} = \varnothing \;\&\; \text{signs} \cap \text{features} = \varnothing$$

3.1.5. "*Observations*" is the set of all the signs, events and features.

$$\text{observations} \equiv \text{signs} \cup \text{events} \cup \text{features}$$

3.1.6. "*Sets of values*" is the set of all the admissible sets of scalar values.

$$\text{sets of values} \equiv \{\}\mathbb{N} \setminus \{\varnothing\}$$

3.1.7. The values do not coincide with the names of observations.

$$\text{observations} \cap (\cup \text{(set: sets of values) set}) = \varnothing$$

3.1.8. "*Possible values*" is a function that takes an observation and returns its possible value range.

$$\text{sort possible values: observation} \to \text{sets of values}$$

3.1.9. Every observation has no less than two values.

$$\text{(observation: } observations\text{)}\ \mu(possible\ values(\text{observation})) \geq 2$$

3.1.10. "*Conditions*" is the set of all possible conditions. It is the set of the sets consisting of structural values. Every condition is a finite set of structural values. Every such a structural value has attributes which are called *feature* и *range of values*. The value of the first one is the name of a feature, and the value of the second one is a proper subset of the possible values of this feature. The empty set represents the identically true condition.

$$\text{conditions} \equiv \{\}\{(\text{condition: (feature} \to \text{features,} \quad \text{range of values} \to \text{sets of values})$$

$$\text{range of values(condition)} \subset \text{possible values(feature(condition))}\}$$

3.1.11. "*Necessary condition*" is a function that takes a sign and returns a condition that is necessary so that the sign can be examined in the situation.

$$\text{sort necessary condition: signs} \to \text{conditions}$$

3.1.12. "*Diseases*" is a class of concepts corresponding to the diseases which have their descriptions in knowledge. Knowledge must contain the description of at least one disease.

$$\text{sort } diseases\text{: } \{\}\mathbb{N} \setminus \{\varnothing\}$$

3.1.13. Every term from the *diseases* is a structural value with three attributes. They are *number of development periods*, *development periods* and *necessary condition*. The value of the first one is a positive integer, the value of the second one is a function that takes the number of a development period and returns an *interval*, and the value of the third attribute is a condition that is necessary so that this disease took place in the situation (if the value of this attribute is the empty set then the condition is considered to be true).

(disease: *diseases*) sort disease:

$$(necessary\ condition \to conditions,\ number\ of\ development\ periods \to I[1, \infty),$$

development periods → (I[1, *number of development periods*] → interval))

3.1.14. "*Dynamics periods*" is a set of structural values with two attributes. They are *duration* and *range of effect*. The value of the first one is an interval, and the value of the second one is a set of values.

dynamics periods ≡ (*duration* → interval, *range of effect* → sets of values)

3.1.15. "*Interval*" is a set of structural values with two attributes. They are *lower bound* and *upper bound*. Their values are positive integers which are minimal and maximal durations of the interval. The duration is measured in positive integers, and the lower bound is less than the upper bound.

interval ≡ (*lower bound* → I[1, ∞), *upper bound* → I[*lower bound* + 1, ∞))

3.2. The basic concepts and ontological agreements which define the reality of the domain (the unknowns of the metaontology model)

Reality in medical diagnostics is considered as the set of the situations corresponding to diagnostic cases (patients). In this section the basic concepts for situation's descriptions and the restrictions on these values are introduced.

3.2.1. "*Moments*" is a function that takes a sign or an event and returns a set of nonnegative integers which are time moments in a situation when the sign was examined or the event happened. Every number means the amount of hours from the beginning of the examination to the moment when the sign was examined or the event happened. If for a sign the value of this function is the empty set then the sign was not examined for the patient. If the same takes place for an event then the event did not happen in the situation.

sort *moments*: *signs* ∪ *events* → {}I [0, ∞)

3.2.2. Every term from the class of *signs* is a function that takes a time moment of examining this sign and returns the value of this sign at this moment in the situation. Every term from the class of *events* is a function that takes a time moment when the event happened and returns the value of this event at this moment in the situation.

(sign or event: *signs* ∪ *events*) sort sign or event: *moments*(sign or event) → *possible values*(sign or event)

3.2.3. "*Examined features*" is the set of features which were examined in the situation.

sort examined features: {}*features*

3.2.4. Every term from the class of *examined features* has the value that this feature has in the situation.

(feature: *examined features*) sort feature: *possible values*(feature)

3.2.5. "*Fulfilled*" is a predicate that takes an element of the set of *conditions* and returns *truth* if and only if for every component of this element which is an examined feature the value of the first attribute of the structural value (*feature*) belongs to the value of the second attribute (*range of values*) in the situation. The empty condition is identically *true*.

fulfilled ≡ (λ (condition: *conditions*) condition ≠ ∅ ⇒ (& (component: condition) *feature*(component) ∈

∈ *examined features* ⇒ j(*feature*(component)) ∈ *range of values*(component)))

3.2.6. If in the situation a sign was examined at least once then the necessary condition of this sign must be fulfilled.

(sign: *signs*) *moments*(sign) ≠ ∅ ⇒ *fulfilled*(*necessary condition*(sign))

3.2.7. "*Diagnosis*" is the set of diseases with which a patient is ill. If a patient is healthy his or her diagnosis is the empty set.

sort diagnosis: {}*diseases*

3.2.8. If a disease belongs to the patient's diagnosis then the necessary condition for this disease must be fulfilled.

(disease: *diagnosis*) *fulfilled*(*necessary condition*(disease))

3.2.9. "*Development*" is a function that takes a disease from the diagnosis or an examined sign. In the first case this function returns a partition of the time axis. Every interval of this partition corresponds to a development period of this disease. In the second case the function also returns a partition of the time axis. During each interval of this partition the values of the sign are determined in the situation by a common cause, which is associated with this interval.

sort *development*: *diagnosis* $\cup$ {(sign: *signs*) *moments*(sign) $\neq \varnothing$} $\rightarrow$ *partitions*

3.2.10. The interval during which the development of a sign is observed covers all the examination moments of this sign.

(sign: *signs*) *moments*(sign) $\neq \varnothing \Rightarrow$ *element*(*development*(sign), 0) $\leq$ inf(*moments*(sign)) &

& *element*(*development*(sign),*length*(*development*(sign))) $\geq$ *sup*(*moments*(sign))

3.2.11. "*Development intervals of sign*" is a set of structural values that consist of two attributes. They are *sign* and *number of interval*. The value of the first attribute is the name of sign and the value of the second one is the number of its development interval.

development intervals of sign $\equiv$ (*sign* $\rightarrow$ *signs*, *number of interval* $\rightarrow$ I[1,*length*(*development*(*sign*))–1])

3.2.12. If a disease belongs to the diagnosis then the number of development periods of this disease in the situation is the same as the number of its development periods in the knowledge base. The duration of each development period in the situation is between the lower and upper bounds for the duration of this development period.

(disease: *diagnosis*) *length*(*development*(disease)) = *number of development periods*(disease)+1 &
& (& (number of a development period: I[1, *length*(*development*(disease)) – 1])
element(*development*(disease), number of a development period) –
– *element*(*development*(disease), number of a development period – 1) $\in$
$\in$ I[*lower bound*(*development periods*(disease)(number of a development period),
upper bound(*development periods*(disease)(number of a development period)])

Conclusion

In this article an informal description of a metaontology of medical diagnostics of acute diseases has been presented. In this metaontology interaction of cause-and-effect relations of different types are taken into account. This metaontology is close to real ideas of the medical diagnostics in the Russian Federation. It defines combined and complicated pathology, the development of pathological processes in time and also the influence of treatment and other events on the manifestation of diseases. In addition, a part of the metaontology model is presented. This part includes the definitions terms of the knowledge model (parameters), the definitions of terms of the reality model (unknowns) and the unenriched logical relationships system consisting of the integrity restrictions for unknowns and parameters.

Bibliography

1. Heijst G.V., Schreiber A.Th., Wielinga B.J. Using explicit ontologies in KBS development // International Journal of Human-Computer Studies. – 1997. – vol.46. – P. 183-292.
2. Waiss S.M., Kulikowski C.A., Amarel S., Safir A. A model-based method for computer-aided medical decision-making // Artificial Intelligence. – 1978. – vol.11, №2. – P. 145 -172.
3. Клещев А.С., Самсонов В.В., Черняховская М.Ю. Медицинская экспертная система Консультант-2. Представление знаний. Препринт. – Владивосток: ИАПУ ДВО АН СССР, 1987. – 44 с.
4. Клещев А.С., Черняховская М.Ю. Медицинские системы-консультанты // Представление знаний в человеко-машинных и робототехнических системах. Том С. «Прикладные человеко-машинные системы, ориентированные на знания.». – М.: ВЦ АН СССР, ВИНИТИ, 1984. – С. 282-309.

5. Клещев А.С., Черняховская М.Ю. Системы представления проблемно-ориентированных знаний. // Техническая кибернетика. – 1982. – № 5. – С. 43-63.
6. Patil R.S. Causal representation of patient illness for electrolyte and acid-base diagnosis. // PhD thesis. Laboratory for Computer Science, MIT, 1981.
7. Черняховская М.Ю. Представление знаний в экспертных системах медицинской диагностики. Владивосток, ДВНЦ АН СССР, 1983. 212 с.
8. Gruber T.R. Ontolingua: A mechanism to support portable ontologies. // Technical report KSL-91-66. Stanford University, Knowledge System Laboratory, Revision, 1992.
9. Genesereth M.R., Fikes R.E. Knowledge intercharge format, version 3.0 reference manual. // Technical Report Logic 92-1, Computer Science Department, Stafford University, 06/1991.
10. Kleshchev A., Artemjeva I. A mathematical apparatus for ontology simulation. An extendable language of applied logic. // International Journal of Information Theories and Applications. – 2005. – vol. 12. - N. 2, P. 149-157.
11. Каменев А.В., Клещев А.С., Черняховская М.Ю. Логическая модель взаимодействия причинно-следственных отношений различных типов в области медицинской диагностики // Препринт. – Владивосток: ИАПУ ДВО РАН, 1999. 56с. http://www.iacp.dvo.ru/es/publ/128.pdf
12. Kleshchev A.S., Artemjeva I.L. Domain ontologies and their mathematical models // In the Proceedings of the XII-th International Conference "Knowledge-Dialog-Solution" - KDS 2006, June 20-25, Varna, Bulgaria, Sofia: FOI-COMMERGE-2006. PP. 107-115. - ISSN 954-16-0038-7.

Authors' Information

Chernyakhovskaya M.Yu. - chernyah@iacp.dvo.ru

Kleschev A.S. - kleschev@iacp.dvo.ru

Moskalenko F.M. - philipmm@yahoo.com

Institute for Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio st., Vladivostok, Russia.

A METAONTOLOGY FOR MEDICAL DIAGNOSTICS OF ACUTE DISEASES. PART 2. A FORMAL DESCRIPTION OF CAUSE-AND-EFFECT RELATIONS

Mary Chernyakhovskaya, Alexander Kleshchev, Phillip Moskalenko

***Abstract**: This article is the continuation of the formal description of the metaontology for medical diagnostics in the language of applied logic. It contains a description of interrelations between terms of knowledge and reality in the form of ontological agreements.*

***Keywords**: Medical Diagnostics, ontology model, metaontology.*

***ACM Classification Keywords**: I.2.1 Applications and Expert Systems, I.2.4 Knowledge Representation Formalisms and Methods, J.3 Life and Medical Sciences.*

***Conference**: The paper is selected from International Conference "Intelligent Information and Engineering Systems" INFOS 2008, Varna, Bulgaria, June-July 2008*

Introduction

This article is the continuation of [1] and contains a description of all the classes of cause-and-effect relations which take place in the situations, a description of knowledge about them, and also ontological agreements on their correspondence.

This paper was made according to the project of RFBR № 06-07 89071 «An investigation of possibilities for collective managing information resources of various levels of generality in the semantic Web » and to the project of FEBRAS № 06-III-A-01-457 «Designing, implementing and developing the bank of medical knowledge in the Internet network».

1. The terms of knowledge and reality which describe normal reactions and ontological agreements on correspondence between them

1.1. "*Knowledge about normal reactions*" is a set of structural values with attributes *effect*, *variants and acting factors.* Each of these values is knowledge about a particular normal reaction. The value of the first attribute is the name of a sign, the value of the second one is a set of variants of the normal reaction for this sign, and the value of the third one is a set of features.

knowledge about normal reactions ≡ (*effect*→*signs*, *variants*→{}*variants of norm*, *acting factors*→{}*features*)

1.2. "*Variants of norm*" is a term of knowledge. It is a set of structural values with attributes *range of effect* and *condition on acting factors*. Every value is knowledge about a particular variant of the normal reaction. The value of the first attribute is the set of values of the sign in this variant, and the value of the second one is a condition.

variants of norm ≡ (*range of effect*→*sets of values*, *condition on acting factors*→*conditions*)

1.3. For any variant of norm the range of the effect is a proper subset of possible values of the sign that is the effect of this normal reaction.

(knowledge: *knowledge about normal reactions*) (variant: *variants*(knowledge))

range of effect(variant) ⊂ *possible values*(*effect*(knowledge))

1.4. "*Normal reactions*" is a term of reality. It is a set of structural values with attributes *effect* and *variant.* Each of these values is a normal reaction that takes place in a situation. The value of the first attribute is the name of a sign, and the value of the second one is a variant of norm.

normal reactions ≡ (*effect*→*signs*, *variant*→ *variants of norm*)

1.5. If in a situation there is a normal reaction then in the set of *knowledge about normal reactions* there is such an element that its effect coincides with the effect of the normal reaction, the variant of norm for this normal reaction belongs to the set of variants of norm for this element, and for this variant of norm the condition on acting factors is fulfilled.

(reaction: *normal reactions*) ($\vee$ (knowledge: *knowledge about normal reactions*)

effect(knowledge) = *effect*(reaction)&*variant*(reaction) $\in$ *variants*(knowledge)&

&*fulfilled*(*condition on acting factors*(*variant*(reaction))))

2. The terms of knowledge and reality which describe responses to event's influence, and ontological agreements on correspondence between them

2.1. "*Knowledge about responses to event's influences*" is a set of structural values with attributes *cause-event*, *effect*, *variants*, *acting factors*, *necessary condition* and *modality*. Every value is knowledge about a particular response to event's influence. The value of the cause-event is the name of an event, the value of the effect is the name of a sign, the value of variants is a set of rei-variants, the value of the acting factors is a set of features, the value of the necessary condition is a condition, and the value of modality (hereafter) is *necessity* or *possibility*.

knowledge about responses to event's influences $\equiv$ (*cause-event*$\rightarrow$*events*, *effect*$\rightarrow$*signs*, *variants*$\rightarrow${}*rei-variants*, *acting factors*$\rightarrow${}*features, necessary condition*$\rightarrow$*conditions*, *modality*$\rightarrow${*possibility*, *necessity*})

2.2. "*Rei-variants*" is a set of structural values with attributes *range of cause-event*, *number of dynamics periods*, *description of dynamics* and *condition on acting factors*. Every value is knowledge about a particular variant of a response to event's influence. The value of the first attribute is a set of event's values. The value of the second one is a positive integer. The value of the third one is a function that takes the number of a dynamics period and returns the dynamics period. The value of the forth attribute is a condition.

rei-variants $\equiv$ (*range of cause-event*$\rightarrow$*sets of values*, *number of dynamics periods*$\rightarrow$I[1, ∞), *description of dynamics*$\rightarrow$(I[1, *number of dynamics periods*]$\rightarrow$*dynamics periods*), *condition on acting factors* $\rightarrow$ *conditions*)

2.3. "*Responses to event's influence*" is a term of reality. It is a set of structural values with attributes *cause-event*, *effect*, *variant*, *dynamics of values* and *modality*. Every value is a response to an event's influence that takes place in the situation. The value of the cause-event is the name of an event, the value of the effect is the name of a sign, the value of the variant is a rei-variant, and the value of the dynamics of values is a partition.

responses to event's influence $\equiv$ (*cause-event*$\rightarrow$*events*, *effect*$\rightarrow$*signs*, *variant*$\rightarrow$*rei-variants*, *dynamics of values*$\rightarrow$*partitions*, *modality*$\rightarrow${*possibility*, *necessity*})

2.4. If in a situation there is a response to an event's influence then the beginning of its dynamics of values belongs to the set of the time moments when the cause-event took place.

(response: *responses to event's influences*)

element(*dynamics of values*(response), 0) $\in$ *moments*(*cause-event*(response))

2.5. If in a situation there is *a response to an event's influence*, and this event happens to the patient at a moment then an element of the set of *knowledge about responses to event's influence* belongs to the model of knowledge. For this element the cause-event is the event, the effect is the same as the effect of the response to the event's influence, the necessary condition is fulfilled, the modality is the same as the modality of the response to the event's influence, the rei-variant of the response to the event's influence belongs to the set of rei-variants, and the value of the cause-event of this rei-variant belongs to the range of the cause-event of this variant, the number of intervals for the dynamics of values of the response to the event's influence is equal to the number of dynamics periods of this variant, and the condition on the acting factors is fulfilled.

(response: *responses to event's influences*)($\vee$ (knowledge: *knowledge about responses to event's influences*)

cause-event(knowledge)=*cause-event*(response)& *effect*(knowledge) = *effect*(response) &

& *fulfilled*(*necessary condition*(knowledge)) & *modality*(knowledge) = *modality*(response)&

& *variant*(response) ∈ *variants*(knowledge)&*cause-event*(response)(*element*(*dynamics of values*(response), 0))∈ *range of cause-event*(*variant*(response)) & *length*(*dynamics of values*(response)) – 1 =

= *number of dynamics periods*(*variant*(response)) & *fulfilled*(*condition on acting factors*(*variant*(response))))

3. The terms of knowledge and reality which describe clinical manifestations of diseases, and ontological agreements on correspondence between them

3.1. "Knowledge about clinical manifestations" is a set of structural values with attributes cause, development period of disease, effect, variants, acting factors, necessary condition and modality. Every value is knowledge about a particular clinical manifestation of a disease. The value of the cause is the name of a disease, the value of the development period of the disease is the number of a development period of the disease, the value of the effect is the name of a sign, the value of the variants is a set of cm-variants, the value of the acting factors is a set of features, the value of the necessary condition is a condition.

knowledge about clinical manifestations ≡ (*cause* → *diseases, development period of disease* →

→ I[1, *number of development periods*(*cause*)], e*ffect* → *signs*, *variants* → {}*cm-variants*,

acting factors → {}*features*, *necessary condition* → *conditions*, *modality* → {*possibility*, *necessity*})

3.2. "*Cm-variants*" is a set of structural values with attributes *number of dynamics periods*, *description of dynamics* and *condition on acting factors*. Every value is knowledge about a particular variant of a clinical manifestation of a disease. The value of the first attribute is a positive integer. The value of the second attribute is a function that takes the number of a dynamics period and returns the dynamics period. The value of the third attribute is a condition.

cm-variants ≡ (*condition on acting factors*→*conditions*, *number of dynamics periods*→I[1, ∞),

description of dynamics→ (I[1, *number of dynamics periods*]→*dynamics periods*))

3.3. "*Clinical manifestations*" is a term of reality. It is a set of structural values with attributes *cause*, *development period of disease*, *effect*, *variant*, *dynamics of values* and *modality.* Every value is a clinical manifestation of a disease which took place in a situation. The value of the cause is the name of a disease from the diagnosis, the value of the development period of the disease is the number of a development period of the disease, the value of the effect is the name of a sign, the value of the variant is a cm-variant, the value of the dynamics of values is a partition.

clinical manifestation ≡ (*cause* → *diagnosis*, *development period of disease* →

→ I[1, *number of development periods*(*cause*)], *effect* → *signs*, *variant* → *cm-variants*,

dynamics of values→*partitions*, *modality*→{*possibility*, *necessity*})

3.4. If in a situation there is a clinical manifestation of a disease from the patient's diagnosis during a development period of this disease then the beginning of its dynamics of value is the same as the moment of the beginning of the development period of the disease, and its end is the same as the end of this period.

(manifestation: *clinical manifestations*) *element*(*dynamics of values*(manifestation), 0) =

= *element*(*development*(*cause*(manifestation)),*development period of disease*(manifestation) –1) &

& *element*(*dynamics of values*(manifestation), *length*(*dynamics of values*(manifestation))) =

= *element*(*development*(*cause*(manifestation)), *development period of disease*(manifestation))

3.5. If in a situation there is a clinical manifestation of a disease from the patient's diagnosis during a development period of this disease then the model of knowledge contains such an element of the set *knowledge about clinical manifestations* for which the following takes place: its cause is the same disease with the same development period, its effect coincides with the effect of the clinical manifestation, its necessary condition is fulfilled, its modality coincides with the modality of the clinical manifestation, and the cm-variant of the clinical

manifestation for which the number of intervals in the dynamics of values is equal to the number of dynamics periods for this variant and the condition on the acting factors is fulfilled belongs to the set of cm-variants.

(manifestation: *clinical manifestations*) ($\vee$ (knowledge: *knowledge about clinical manifestations*)

cause(knowledge) = *cause*(manifestation) & *development period of disease*(knowledge) =

= *development period of disease*(manifestation) & *effect*(knowledge) = *effect*(manifestation) &

& *fulfilled*(*necessary condition*(knowledge)) & *modality*(knowledge) = *modality*(manifestation) &

& *variant*(manifestation) $\in$ *variants*(knowledge) & *length*(*dynamics of values*(manifestation)) – 1 = *number of dynamics periods*(*variant*(manifestation)) & *fulfilled*(*condition on acting factors*(*variant*(manifestation))))

4. The terms of knowledge and reality which describe clinical manifestations modified by event's influence and ontological agreements on correspondence between them

4.1. "Knowledge about clinical manifestations modified by event's influence" is a set of structural values with attributes cause, cause-event, effect, variants, acting factors, necessary condition and modality. Every value is knowledge about a particular clinical manifestation modified by event's influence. The value of the cause is the name of a disease, the value of the cause-event is the name of an event, the value of the effect is the name of a sign, the value of the variants is a set of cmmei-variants, the value of the acting factors is a set of features, and the value of the necessary condition is a condition.

knowledge about clinical manifestations modified by event's influence $\equiv$ (*cause* $\rightarrow$ *diseases*, *cause-event* $\rightarrow$

$\rightarrow$ *events*, *effect* $\rightarrow$ *signs*, *variants* $\rightarrow$ {}*cmmei-variants*, *acting factors* $\rightarrow$ {}*features*, *necessary condition* $\rightarrow$

$\rightarrow$ *conditions*, *modality* $\rightarrow$ {*possibility*, *necessity*})

4.2. "*Cmmei-variants*" is a set of structural values with attributes *range of cause-event*, *number of dynamics periods*, *description of dynamics* and *condition on acting factors*. Every value is knowledge about a particular variant of a clinical manifestation modified by event's influence. The value of the first attribute is a set of values of the event, the value of the second one is a positive integer, the value of the third one is a function that takes the number of a dynamics period and returns the dynamics period, the value of the forth one is a condition.

cmmei-variants $\equiv$ (*range of cause-event* $\rightarrow$ *sets of values*, *number of dynamics periods* $\rightarrow$ I[1, ∞),

description of dynamics $\rightarrow$ (I[1, *number of dynamics periods*] $\rightarrow$ *dynamics periods*),

condition on acting factors $\rightarrow$ *conditions*)

4.3. If knowledge contains the definition of a clinical manifestation of a disease modified by an event's influence then for all the development periods of the disease must be defined a clinical manifestation having the same sign as the effect.

(knowledge1: *knowledge about clinical manifestations modified by event's influence*)

(number: I[1, *number of development periods*(*cause*(knowledge))])

($\vee$ (knowledge2: *knowledge about clinical manifestations*)

cause(knowledge2) = *cause*(knowledge1) & *effect*(knowledge2) = *effect*(knowledge1) &

& *development period*(knowledge2) = number)

4.4. "*Clinical manifestations modified by event's influence*" is a term of reality. It is a set of structural values with attributes *cause*, *cause-event*, *effect*, *variant*, *dynamics of values* and *modality*. Every value is a clinical manifestation modified by an event's influence which takes place in a situation. The value of the cause is the name of a disease from the patient's diagnoses, the value of the cause-event is the name of an event, the value of the effect is the name of a sign, the value of the variant is a cmmei-variant, and the value of the dynamics of values is a partition.

clinical manifestations modified by event's influence $\equiv$ (*cause* $\rightarrow$ *diagnosis*, *cause-event* $\rightarrow$ *events*, *effect* $\rightarrow$

→ *signs*, *variant* → *cmmei-variants*, *dynamics of values* → *partitions, modality* → {*possibility*, *necessity*})

4.5. If in a situation there is a clinical manifestation of a disease from the patient's diagnosis modified by an event's influence then the beginning of its dynamics of values belongs to time moments when the cause-event happened.

(manifestation: *clinical manifestations modified by event's influence*)

element(*dynamics of values*(manifestation), 0) ∈ *moments*(*cause event*(manifestation))

4.6. If in a situation there is a clinical manifestation of a disease from the patient's diagnosis modified by an event's influence and this event happened to the patient at a time moment then the model of knowledge contains such an element of the set *knowledge about clinical manifestations modified by event's influence* for which the following takes place: its cause is this disease, its cause-event is this event, its effect is the same as the effect of the clinical manifestation modified by the event's influence, its necessary condition is fulfilled, its modality is the same as the modality of the clinical manifestation modified by the event's influence, and its set of cmmei-variants contains the cmmei-variant of the clinical manifestation modified by the event's influence and the value of the cause-event of this cmmei-variant belongs to the range of the cause-event for this variant, the number of intervals of the dynamics of values for the clinical manifestation modified by the event's influence is equal to the number of dynamics periods for this variant, and the condition on acting factors is fulfilled.

(manifestation: *clinical manifestations modified by event's influence*)

(∨ (knowledge: *knowledge about clinical manifestations modified by event's influence*)

cause(knowledge) = *cause*(manifestation) & *cause-event*(knowledge) = *cause-event*(manifestation) &

& *effect*(knowledge) = *effect*(manifestation) & *fulfilled*(*necessary condition*(knowledge)) &

& *modality*(knowledge) = *modality*(manifestation) & *variant*(manifestation) ∈ *variants*(knowledge) &

& *cause-event*(manifestation)(*element*(*dynamics of values*(manifestation), 0)) ∈ *range of cause-event*(*variant*(

manifestation)) & *length*(*dynamics of values*(manifestation)) – 1 = *number of dynamics periods*(*variant*(

manifestation)) & *fulfilled*(*condition on acting factors*(*variant*(manifestation))))

5. The terms of knowledge and reality which describe etiologies and ontological agreements on correspondence between them

5.1. "*Knowledge about etiologies*" is a set of structural values with attributes *cause-event*, *effect*, *variants*, *modality*, *necessary condition* and *acting factors*. Every value describes knowledge about a particular etiology. The value of the cause-event is the name of an event, the value of the effect is the name of a disease, the value of the variants is a set of variants of etiology, the value of the acting factors is a set of features, the value of the necessary condition is a condition, the value of the modality is *necessity* or *possibility*.

knowledge about etiologies ≡ (*cause-event* → *events*, *effect* → *diseases*, *variants* → {}*variants of etiology*,

acting factors → {}*features*, *necessary condition* → *conditions*, *modality* → {*possibility*, *necessity*})

5.2. "*Variants of etiology*" is a set of structural values with attributes *range of cause-event*, *description of dynamics*, and *condition on action factors*. Every value describes knowledge about a particular variant of etiology. The value of the first attribute is a set of values of the event, the value of the second one is an interval, and the value of the third one is a condition.

variants of etiology ≡ (*range of cause-event* → *sets of values*, *description of dynamics* → *interval*,

condition on acting factors → *conditions*)

5.3. "*Etiologies*" is a term of reality. It is a set of structural values with attributes *cause-event*, *moment*, *effect*, *variant* and *modality*. Every value describes the etiology that takes place in a situation. The value of the cause-event is the name of an event, the value of the moment is the time moment when the event happened, the value

of the effect is the name of a disease from the patient's diagnosis, the value of the variant is a variant of the etiology, and the value of the modality is *necessity* or *possibility*.

etiologies ≡ (*cause-event* → *events*, *moment* → *moments*(*cause-event*), *effect* → *diagnosis*, *variant* → *variants of etiology*, *modality* → {*possibility*, *necessity*})

5.4. If in a situation there is the etiology caused by an event that happened to the patient at a time moment and the effect of the etiology is a disease from the patient's diagnosis then the model of knowledge contains an element of the set *knowledge about etiology* for which the following takes place:

- its cause-event is this event,
- its effect is this disease,
- its necessary condition is fulfilled,
- its modality is the same as the modality of the etiology,
- its set of variants of etiology contains the variant of this etiology for which: the value of the cause-event belongs to the range of the cause-event of this variant; the duration of the interval between the moment when the cause-event happened and the beginning of the disease is included between the lower and upper bounds of the interval from the dynamics description of this variant; its condition on acting factors is fulfilled.

(etiology: *etiologies*) (∨ (knowledge about etiology: *knowledge about etiologies*)
cause-event(knowledge about etiology) = *cause-event*(etiology) & *effect*(knowledge about etiology) = *effect*(etiology) & *fulfilled*(*necessary condition*(knowledge about etiology)) & *modality*(knowledge about etiology) = = *modality*(etiology) & *variant*(etiology) ∈ *variants*(knowledge about etiology) & *cause-event*(etiology) (*moment*(etiology)) ∈ *range of cause-event*(*variant*(etiology)) & *moment*(etiology) – *element*(*development*(*effect*(etiology)), 0)∈ I[*lower bound*(*description of dynamics* (*variant*(etiology)), *upper bound*(*description of dynamics*(*variant*(etiology))]) & *fulfilled*(*condition on acting factors*(*variant*(etiology))))

6. The terms of knowledge and reality which describe complications and ontological agreements on correspondence between them

6.1. "*Knowledge about complications*" is a set of structural values with attributes *cause*, *effect*, *variants*, *acting factors*, *necessary condition* and *modality*. Every value describes knowledge about a particular complication. The values of the cause and effect are diseases, the value of variants is a set of variants of a complication, the value of the acting factors is a set of features, the value of the necessary condition is a condition, and the value of the modality is *necessity* or *possibility*.

knowledge about complications ≡ (*cause* → *diseases*, *effect* → *diseases*, *variants* → {}*variants of complication*, *acting factors* → *{}features, necessary condition* → *conditions, modality* → *{possibility, necessity})*

6.2. "*Variants of complication*" is a set of structural values with attributes *description of dynamics* and *condition on acting factors*. Every value describes knowledge about a particular variant of a complication. The value of the first attribute is an interval, and the value of the second one is a condition.

variants of complication ≡ (*description of dynamics* → *interval*, *condition on acting factors* → *conditions*)

6.3. "*Complications*" is a term of reality. It is a set of structural values with attributes *cause*, *effect*, *variant* and *modality*. Every value describes a complication of a disease by another one which takes place in a situation. The values of the cause and effect are diseases from the diagnosis, the value of the variant is a variant of the complication, and the value of the modality is *necessity* or *possibility*.

complication ≡ (*cause* → *diagnosis*, *effect* → *diagnosis*, *variant* → *variants of complication*, *modality* → {*possibility*, *necessity*})

6.4. If in a situation there is a complication and its effect and cause are diseases from the diagnosis then the model of knowledge contains an element of the set *knowledge about complications for which the following takes place*:

- its cause is the disease that is the cause of the complication,
- its effect is the disease that is the effect of the complication,
- its necessary condition is fulfilled,
- its modality is the same as the modality of the complication,
- its set of variants of complication contains the variant of the considered complication and the duration of the interval between the beginning of the disease-cause and the beginning of the disease-effect is included between the lower and upper bounds of the duration of the interval from the description of dynamics of this variant; its condition on acting factors is fulfilled.

(complication: *complications*) ($\vee$ (knowledge1: *knowledge about complications*)
cause(knowledge1) = *cause*(complication) & e*ffect*(knowledge1) = *effect*(complication) & *fulfilled*(*necessary condition*(knowledge1)) & *modality*(knowledge1) = *modality*(complication) & *variant*(complication) $\in$ *variants*(knowledge1)&*element*(*development*(*effect*(complication)),0)–*element*(*development*(*cause*(complication)),0)$\in$ $\in$I[*lower bound*(*description of dynamics*(*variant*(complication)), *upper bound*(*description of dynamics*(*variant*(complication))]) & *fulfilled*(*condition on acting factors*(*variant*(complication))))

6.5. "*Connection*" is the predicate that corresponds to the transitive closure of the relation *complication*.

connection $\equiv$ (λ (disease1: *diseases*) (disease2: *diseases*)
($\vee$ (knowledge about complication: *knowledge about complications*)
cause(knowledge about complication)= disease1 & *effect*(knowledge about complication)= disease2) $\vee$
$\vee$ ($\vee$ (disease: *diseases*) *connection*(disease1, disease) & *connection*(disease, disease2)))

6.6. A disease can be its complication neither directly nor indirectly.

(disease: *diseases*) $\neg$ *connection*(disease, disease)

7. General terms and ontological agreements which are used for describing cause-and effect relations

7.1. A feature that is a part of the condition on acting factors of a variant of a cause-and-effect relation is an acting factor of this relation.

(knowledge1: *knowledge about normal reactions* $\cup$ *knowledge about responses to event's influence* $\cup$
$\cup$ *knowledge about clinical manifestations* $\cup$ *knowledge about clinical manifestations modified by event's influence* $\cup$ *knowledge about etiologies* $\cup$ *knowledge about complications*)
(variant: *variants*(knowledge1)) (CAF: *condition on acting factors*(variant))
feature(CAF) $\in$ *acting factors*(knowledge1)

7.2. The range of the cause-event for any element of the sets *knowledge about etiologies, knowledge about responses to event's influence*, *knowledge about clinical manifestations modified by event's influence* for any its variant is a proper subset of possible values of the event that is the cause-event.

(knowledge1: *knowledge about responses to event's influence* $\cup$ *knowledge about clinical manifestations modified by event's influence* $\cup$ *knowledge about etiologies*) (variant: *variants*(knowledge1))
range of cause-event(variant) $\subset$ *possible values*(*cause-event*(knowledge1))

7.3. For every element of the sets *knowledge about clinical manifestations*, *knowledge about clinical manifestations modified by event's influence*, *knowledge about responses to event's influence* for any its variant and for every its dynamics period the range of the effect is a subset of possible values of the sign that is the effect of this element of knowledge.

(knowledge1: *knowledge about responses to event's influence* ∪ *knowledge about clinical manifestations* ∪

∪ *knowledge about clinical manifestations modified by event's influence*)

(variant: *variants*(knowledge1))

(number of dynamics period: I[1, *number of dynamics periods*(variant))

(dynamics period: *description of dynamics*(variant)(number of dynamics period))

range of effect(dynamics period) ⊆ *possible values*(*effect*(knowledge1))

7.4. The auxiliary term *cause-and effect relations* is the set of values for the terms from *normal reactions*, *responses to event's influence*, *clinical manifestations* and *clinical manifestations modified by event's influence*.

cause-and effect relations ≡ *normal reactions* ∪ *responses to event's influence* ∪

∪ *clinical manifestations* ∪ *clinical manifestations modified by event's influence*

7.5. If in a situation there is a *response to event's influence*, *clinical manifestation* or *clinical manifestation modified by event's influence* then the duration of every dynamics period from *dynamics of values* of this cause-and-effect relation belongs to the interval of admissible durations of this dynamics period for the variant of this cause-and effect relation.

(cause-and-effect relation: *cause-and-effect relations* \ *normal reactions*)

(number of dynamics period: I[1, *number of dynamics periods*(*variant*(cause-and-effect relation))])

element(*dynamics of values*(cause-and-effect relation), number of dynamics period) – *element*(*dynamics of values*(cause-and-effect relation), number of dynamics period –1) ∈ I[*lower bound*(*duration*(*description of dynamics*(*variant*(cause-and-effect relation))(number of dynamics period))), *upper bound*(*duration*(*description of dynamics*(*variant* cause-and-effect relation))(number of dynamics period)))]

Conclusion

In this article the next part of the metaontology model for medical diagnostics is presented. This part includes the description of interrelations between knowledge about cause-and-effect relations and these relations in reality.

Bibliography

1. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 1. An informal description and definitions of basic terms. International Book Series "Information Science and Computing" – Book "Algorithmic and Mathematical Foundations of the Artificial intelligence", ITHEA, Sofia, Bulgaria, 2008, pp. 103-111.

Authors' Information

Chernyakhovskaya M.Yu. - chernyah@iacp.dvo.ru

Kleschev A.S. - kleschev@iacp.dvo.ru

Moskalenko F.M. - philipmm@yahoo.com

Institute for Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio st., Vladivostok, Russia.

A METAONTOLOGY FOR MEDICAL DIAGNOSTICS OF ACUTE DISEASES. PART 3. A FORMAL DESCRIPTION OF THE CAUSES OF SIGNS' VALUES AND OF DISEASES

Mary Chernyakhovskaya, Alexander Kleshchev, Phillip Moskalenko

***Abstract**: This article is the final part of the formal description of the metaontology for medical diagnostics in the language of applied logic. It contains a description of the causes of signs' values and of the causes of diseases.*

***Keywords**: Medical Diagnostics, ontology model, metaontology.*

***ACM Classification Keywords**: I.2.1 Applications and Expert Systems, I.2.4 Knowledge Representation Formalisms and Methods, J.3 Life and Medical Sciences.*

***Conference**: The paper is selected from International Conference "Intelligent Information and Engineering Systems" INFOS 2008, Varna, Bulgaria, June-July 2008*

Introduction

This article is the continuation of [1] and [2] where the formal definitions of the basic terms of the metaontology and the terms of knowledge and situations that describe the cause-and-effect relations were given. In this article the terms which describe the causes of values of signs during intervals of their development, and also the causes of diseases from the diagnosis are presented.

This paper was made according to the project of RFBR № 06-07 89071 «An investigation of possibilities for collective managing information resources of various levels of generality in the semantic Web » and to the project of FEBRAS № 06-III-A-01-457 «Designing, implementing and developing the bank of medical knowledge in the Internet network».

1. The terms and agreements which describe development intervals of a sign

In this section the relationships are presented which specify the cause-and-effect relations which act during an interval of the partition of the time axes corresponded to a sign. These relationships determine the values of this sign and also the bounds of the partition.

1.1. A description of the causes of the values of a sign during an interval of its development and their properties

In this section the relationships are presented that specify which from the cause-and-effect relations that act during a time interval associated with an examined sign determine the values of this sign. For this purpose the set of all the possible cause-and-effect regularities which can regulate the values of the considered sign at all the time moments belonging to this interval is defined. A concept of the priority of a cause of the values of a sign is introduced. The priority depends on the modality of a cause-and-effect relation («necessity» has higher priority than «possibility») and on the class to which the relation belongs (the order of priority's increasing is the following: a normal reaction << a response to an event's influence << a clinical manifestation << a clinical manifestation modified by an event's influence << a clinical manifestation of a disease-complication << a clinical manifestation of a disease-complication modified by an event's influence). The cause-and-effect relation which determines the values of the sign during this development interval has the maximal priority for the set of possible causes.

1.1.1. "*Possible causes for values of sign*" is a function that takes a development interval of a sign and returns the set of all such cause-and-effect relations (normal reactions, responses to event's influence, clinical manifestations and clinical manifestations modified by event's influence) from the situation which proceed during the whole

development interval of the sign and thus, can determine the values of this sign during this interval (moreover, if a cause-and-effect relation is a clinical manifestation or a clinical manifestation modified by event's influence then the disease proceeds during the whole development interval of the sign).

possible causes for values of sign ≡ (λ (development interval of sign: development intervals of sign)
{(normal reaction: *normal reactions*) *effect*(normal reaction)=*sign*(development interval of sign)} ∪
∪ {(response to event's influence: *responses to event's influence*)
effect(response to event's influence) = *sign*(development interval of sign) &
& *interval*(*development*(*sign*(development interval of sign)), *number of interval*(development interval of sign)) ⊆ I[*element*(*dynamics of values*(*variant*(response to event's influence)),0),
element(*dynamics of values*(*variant*(response to event's influence)),
length(*dynamics of values*(response to event's influence)))]} ∪
∪ {(cause-and-effect relation: clinical manifestations ∪ clinical manifestations modified by event's influence)
effect(cause-and-effect relation) = *sign*(development interval of sign) &
&*interval*(*development*(*sign*(development interval of sign)),*number of interval*(development interval of sign))⊆
⊆ I[*element*(*dynamics of values*(*variant*(cause-and-effect relation)),0),
element(dynamics of values(variant(cause-and-effect relation)), length(dynamics of values(cause-and-effect relation)))] & *interval*(*development*(*sign*(development interval of sign)), *number of interval*(development interval of sign)) ⊆ I[*element*(*development*(*cause*(cause-and-effect relation)), 0), *element*(*development*(*cause*(cause-and-effect relation)), *length*(*development*(*cause*(cause-and-effect relation)))}

1.1.2. "*Priority of values of sign*" is a predicate of two variables. It takes two possible causes which can determine the values of signs during their development intervals and returns truth if and only if the both causes relate to the same sign and to the same development interval of this sign and the first cause is more priority than the second one, that is one of the following takes place:

a) the cause-and-effect relation corresponding to the value of the first argument is a response to event's influence with modality of *necessity* and the cause-and-effect relation corresponding to the value of the second argument is one of the following:

- a normal reaction,
- a response to event's influence with the modality of *possibility*;

b) the cause-and-effect relation corresponding to the value of the first argument is a clinical manifestation of a disease from the diagnosis with the modality of *necessity* and the cause-and-effect relation corresponding to the value of the second argument is one of the following:

- a normal reaction,
- a response to event's influence,
- a clinical manifestation of a disease from the diagnosis with the modality of *possibility*;

c) the cause-and-effect relation corresponding to the value of the first argument is a clinical manifestation of a disease from the diagnosis modified by an event's influence with the modality of *necessity* and the cause-and-effect relation corresponding to the value of the second argument is one of the following:

- a normal reaction,
- a response to event's influence,
- a clinical manifestation of a disease from the diagnosis,
- a clinical manifestation of a disease from the diagnosis modified by an event's influence with the modality of *possibility*;

d) the cause-and-effect relation corresponding to the value of the first argument is a clinical manifestation of a disease from the diagnosis (possibly modified by an event's influence), and this disease is a complication (possibly indirect) of another disease, its modality is *necessity*, and the cause-and-effect relation corresponding to the value of the second argument is one of the following:

- a clinical manifestation of the second disease,
- a clinical manifestation of the second disease modified by an event's influence.

priority of values of sign ≡ (λ(CER1: cause-and-effect relations) (CER2: cause-and-effect relations)
(∨ (development interval of sign: *development intervals of sign*)
CER1∈*possible causes for values of sign* (development interval of sign)&
&CER2∈ *possible causes for values of sign* (development interval of sign)&
& ((CER1 ∈ responses to event's influence & modality(CER1) = necessity &
& (CER2 ∈ normal reactions ∨ CER2 ∈ responses to event's influence & modality (CER2) = possibility)) ∨
∨ (CER1∈clinical manifestations & modality(CER1) = necessity &
& (CER2 ∈ normal reactions ∪ responses to event's influence ∨ CER2∈ clinical manifestations &
& modality(CER2)=possibility))∨ (CER1 ∈ clinical manifestations modified by event's influence &
& modality(CER1)=necessity & (CER2 ∈ normal reactions ∪ responses to event's influence ∪ clinical
manifestations ∨CER2∈clinical manifestations modified by event's influence &modality(CER2)=possibility)) ∨
∨(CER1 ∈ clinical manifestations ∪ clinical manifestations modified by event's influence & modality(CER1) =
= *necessity* & (∨ ((another disease: *diagnosis*) *connection*(another disease, *cause*(CER1) &
& (CER2 ∈ clinical manifestations ∪ clinical manifestations modified by event's influence & cause(CER2) =
= another disease)))))))

1.1.3. "*Causes for values of sign with maximum priority*" is a function which takes a sign's development interval and returns only those cause-and-effect relations from the set of *possible causes for values of sign* which have the maximum priority.

causes for values of sign with maximum priority ≡
≡ (λ (development interval of sign: *development intervals of sign*)
possible causes for values of sign(development interval of sign) \
\ {(CER1: *possible causes for values of sign*(development interval of sign))
(∨ (CER2: *possible causes for values of sign*(development interval of sign))
priority of values of sign(CER2, CER1))})

1.1.4. *"Cause of values of sign"* is a function that takes a development interval of a sign and returns the cause-and-effect relation that takes place in the situation (a normal reaction; a response to the event's influence that happened at a moment; a clinical manifestation of a disease from the diagnosis during some its development period; a clinical manifestation of a disease from the diagnosis modified by event's influence that happened at a moment) and determines the values of this sign during this development interval.

sort cause of values of sign: development intervals of sign → cause-and-effect relations

1.1.5. The cause of the values of a sign which were obtained during a development interval of this sign has the maximum priority.

(development interval of sign: *development intervals of sign*)
cause of values of sign(development interval of a sign) ∈
∈ causes for values of sign with maximum priority(development interval of a sign)

1.1.6. If the cause of the values of a sign which were obtained during a development interval of this sign is a normal reaction then any value of this sign examined at any moment during this interval belongs to the range of the effect for a variant of this normal reaction.

(development interval of sign: *development intervals of sign*)
(moment of examination: *moments* (*sign*(development interval of sign)) ∩
∩ *interval*(*development*(*sign*(development interval of sign)),
number of interval(development interval of sign)))

cause of values of sign(development interval of sign) ∈ normal reactions ⇒

⇒ *sign*(development interval of sign)(moment of examination) ∈

∈ range of effect(variant(cause of values of sign(development interval of sign)))

1.1.7. If the cause that determines the values of a sign which were obtained during a development interval of this sign is a cause-and-effect relation of one of the following types:

- a response to event's influence;
- a clinical manifestation;
- a clinical manifestation modified by event's influence

then the values of this sign at any moment of the sign's examination that belongs to this development interval of this sign and to a dynamics period of the active variant of this cause-and-effect relation belong to the range of the effect for this dynamics period of this variant of the cause-and-effect relation.

(development interval of sign: {(interval: development intervals of sign) cause of values of sign ∈

∈ cause-and-effect relations \ normal reactions })

(dynamics period:

I[1, number of dynamics periods(variant(cause of values of sign(development interval of sign)))])

(moment of examination: *moments*(*sign*(development interval of sign)) ∩ *interval*(*development*(*sign*(

development interval of sign)), *interval number*(development interval of sign)) ∩ *interval*(

dynamics of values(*cause of values of sign*(development interval of sign)), moment of examination))

sign(development interval of sign)(moment of examination) ∈

∈ range of effect(description of dynamics(variant(cause of values of sign(development interval of sign)))

(dynamics period))

1.2. Properties of a partition of the time axis of a sign

This section contains the conditions for borders of time axis partition intervals related to a sign.

1.2.1. *"The same cause-and-effect relation"* is a predicate which takes two cause-and-effect relations and returns *true* if and only if their effects are the same and furthermore they both have one of the following types:

- normal reactions;
- responses to event's influence with the same cause-event that happened at the same moment;
- clinical manifestations of the same disease from the diagnosis during the same development interval;
- clinical manifestations of the same disease from the diagnosis modified by the same event that happened at the same moment.

the same cause-and-effect relation ≡

≡ (λ (CER1: *cause-and-effect relations*)

(CER2: *cause-and-effect relations*)

(CER1 ∈ normal reactions & CER2 ∈ normal reactions ∨

∨ CER1 ∈ *responses to event's influence* & CER2 ∈ *responses to event's influence* &

& *reason-event*(CER1) = *reason-event*(CER2) &

& *element*(*dynamics*(CER1), 0) = *element*(*dynamics*(CER2), 0) ∨

∨ CER1 ∈ *clinical manifestations* & CER2 ∈ *clinical manifestations* &

& *cause*(CER1) = *cause*(CER2) &

& *development period of disease*(CER1) = *development period of disease* (CER2) ∨

∨ CER1 ∈ *clinical manifestations modified be event's influence* &

& CER2 ∈ *clinical manifestations modified be event's influence* &

& *cause*(CER1) = *cause*(CER2) & *cause-event*(CER1) = *cause-event*(CER2) &

& *element*(*dynamics*(CER1), 0) = *element*(*dynamics*(CER2), 0)) & *effect*(CER1) = *effect*(CER2))

1.2.2. If the values of a sign are determined by the same cause-and-effect relation during two development intervals of this sign then they are determined by the same variant of this relation.

(interval 1: *development intervals of sign*)

(interval 2: *development intervals of sign*)

sign(interval 1) = *sign*(interval 2) &

& *the same cause-end-effect relation*(*cause of values of sign*(interval 1), *cause of values of sign*(interval 2))⇒

⇒ *variant*(*cause of values of sign*(interval 1)) = *variant*(*cause of values of sign*(interval 2))

1.2.3. *"Adjacent development intervals of sign"* is a predicate that takes two development intervals of a sign and returns *true* if and only if the sign has no less than two development intervals and the first interval precedes the second.

adjacent development intervals of sign ≡

≡ (λ (interval 1: *development intervals of sign*)(interval 2: *development intervals of sign*)

sign(interval 1) = *sign*(interval 2) & *number of an interval*(interval 2) – *number of an interval*(interval 1) = 1)

1.2.4. The causes for values of a sign during adjacent development intervals of this sign are different.

(interval 1: *development intervals of sign*) (interval 2: *development intervals of sign*)

adjacent development intervals of sign(interval 1, interval 2) ⇒

⇒ *cause of values of sign*(interval 1) ≠ *cause of values of sign*(interval 2)

1.2.5. If the causes of values of a sign during adjacent development intervals are not normal reactions then the bound between these intervals is either the end of the dynamics of values of the cause-end-effect relation during the first interval or the beginning of the dynamics of values of the cause-end-effect relation during the second interval.

(interval 1: *development intervals of sign*) (interval 2: *development intervals of sign*)

adjacent development intervals of sign(interval 1, interval 2) &

& *cause of values of sign*(interval 1) ∉ *normal reactions* &

& *cause of values of sign*(interval 2) ∉ *normal reactions*) ⇒

⇒ *element*(*development*(*sign*(interval 1)), *interval number*(interval 1)) ∈

∈ {*element*(*dynamics of values*(*cause of values of sign*(interval 1)),

length(*dynamics of values*(*cause of values of sign*(interval 1)))),

element(*dynamics of values*(*cause of values of sign*(interval 2)), 0)}

1.2.6. If the reason of values of a sign during one of its development intervals is a normal reaction then the border between this and next interval is the beginning of the dynamics of values for the cause-and-effect relation which is the cause of the values of the sign during the second interval.

(interval 1: *development intervals of sign*) (interval 2: *development intervals of sign*)

adjacent development intervals of sign(interval 1, interval 2) &

& *cause of values of sign*(interval 1) ∈ *normal reactions* ⇒

⇒ *element*(*development*(*sign*(interval 1)), *interval number*(interval 1))

= *element*(*dynamics of values*(*cause of values of sign*(interval 2)), 0)

1.2.7. If the cause of the values of a sign during one of its development intervals is a normal reaction then the border between this and previous interval is the end of the dynamics of values for the cause-end-effect relation which is the cause of the values of the sign during the first interval.

(interval 1: *development intervals of sign*)

(interval 2: *development intervals of sign*)

adjacent development intervals of sign(interval 1, interval 2) &

& *cause of values of sign*(interval 2) ∈ *normal reactions*) ⇒

$\Rightarrow$ *element*(*development*(*sign*(interval 1)), *interval number*(interval 1)) =
= *element*(*dynamics of values*(*cause of values of sign*(interval 1),
length(*dynamics of values*(*cause of values of sign*(interval 1)))

2. A description of causes of diseases from diagnosis

In this section the relationships are presented which determine what cause-and-effect relation (from the sets of etiologies and complications) is the real cause of a disease from the diagnosis. For this purpose the set of all possible casual relations is determined which might be the cause of the disease. A concept of priority for disease's cause is introduced. The priority depends on the modality of cause-and-effect relation ("*necessity*" has more priority than "*possibility*") and on the moment when this relation begins to act (the earlier cause has more priority). The cause-and-effect relation which is the cause of the disease has the maximum priority among all the possible causes.

2.1. *"Possible causes of disease"* is a function that takes a disease from the patient's diagnosis and returns the set of all the complications and etiologies from the situation for which the effect is this disease.

possible causes of a disease $\equiv$
$\equiv$ (λ(disease: *diagnosis*)
{ (cause-end-effect relation: *complications* $\cup$ *etiologies*)
effect(cause-end-effect relation) = disease})

2.2. "*Priority of disease's causes*" is a predicate of two variables. It takes two possible causes of a disease and returns *true* if and only if both causes are related to the same disease and the first cause has more priority than the second one that means one of the following:

- the modality of each cause-and-effect relation is *necessity* and the first relation started to act earlier than the second one;

- the modality of the first cause-and-effect relation is *necessity*, the modality of the second one is *possibility* and the first one started to act not later than the second one.

priority of disease's causes $\equiv$
$\equiv$ (λ (CER1: *complications* $\cup$ *etiologies*)(CER2: *complications* $\cup$ *etiologies*)
($\vee$ (disease: *diagnosis*)
CER1 $\in$ *possible causes of disease*(disease) &
& CER2 $\in$ *possible causes of disease*(disease) &
& / ((CER1 $\in$ *etiologies*) & (*modality*(CER1) = *necessity*) $\Rightarrow$ *moment*(CER1)),
((CER1 $\in$ *complications*) & (*modality*(CER1) = *necessity*) $\Rightarrow$ *element*(*development*(*cause*(CER1)), 0))) / <
< /((CER2 $\in$ *etiologies*) & (*modality*(CER2) = *necessity*) $\Rightarrow$ *moment*(CER2)),
((CER2 $\in$ *complications*) & (*modality*(CER2) = *necessity*) $\Rightarrow$ *element*(*development*(*cause*(CER2)), 0)))/) $\vee$
$\vee$ /((CER1 $\in$ *etiologies*) & (*modality*(CER1) = *necessity*) $\Rightarrow$ *moment*(CER1)),
((CER1 $\in$ *complications*) & (*modality*(CER1) = *necessity*) $\Rightarrow$ *element*(*development*(*cause*(CER1)), 0)))/ $\leq$
$\leq$ /((CER2 $\in$ *etiologies*) & (*modality*(CER2) = *possibility*) $\Rightarrow$ *moment*(CER2)),
((CER2 $\in$ *complications*) & (*modality*(CER2) = *possibility*) $\Rightarrow$ *element*(*development*(*cause*(CER2)), 0)))/)

2.3. "*Causes of disease with maximum priority*" is a function that takes a disease from the diagnosis and returns the set of all the causes of this disease which have the maximum priority.

causes of disease with maximum priority $\equiv$
$\equiv$ (λ (disease: *diagnosis*) *possible causes of disease*(disease)
\ {(CER1: *possible causes of disease*(disease))
($\vee$ (CER2: *possible causes of disease*(disease))
priority of causes of disease (CER2, CER1))})

2.4. "*Cause of disease*" is a function that takes a disease from the diagnosis and returns its cause. The cause of a disease can be either an etiology or a complication.

sort *cause of disease*: *diagnosis* $\rightarrow$ *etiologies* $\cup$ *complications*

2.5. The cause of a disease from the diagnosis has the maximum priority.

(disease: *diagnosis*) *cause of disease*(disease) $\in$ causes of disease with maximum priority(disease)

Conclusion

In this article the final part of the metaontology model for medical diagnostics has been presented. This model describes the interrelation of cause-and-effect relations of different types. The metaontology is close to practical concepts of medicine in the Russian Federation and describes the combined and complicated pathology, the dynamics of pathological processes in time and the influence of medical treatment and other events on the manifestation of diseases. The model of metaontology includes the definitions of terms of the knowledge model (parameters), definitions of the terms of the situation model (unknowns), and also a system of relationships that consists of integrity constraints for unknowns and parameters and of relationships between them.

The relationships between unknowns and parameters can be divided into the following groups:

1) the relationships between knowledge about cause-and-effect relations and cause-and-effect relations which take place in situations;

2) the relationships which determine cause-and-effect relations that are the reasons of values of each sign during its development intervals;

3) the relationships which determine the properties of borders of intervals of the time axis for each sign;

4) the relationships which determine the reason of each disease from the diagnosis.

Bibliography

1. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 1. An informal description and definitions of basic terms. International Book Series "Information Science and Computing" – Book "Algorithmic and Mathematical Foundations of the Artificial intelligence", ITHEA, Sofia, Bulgaria, 2008, pp. 103-111.
2. Chernyakhovskaya M.Yu., Kleschev A.S., Moskalenko F.M. A metaontology for medical diagnostics of acute diseases. Part 2. A formal description of cause-and-effect relations. International Book Series "Information Science and Computing" – Book "Algorithmic and Mathematical Foundations of the Artificial intelligence", ITHEA, Sofia, Bulgaria, 2008, pp. 112-119.

Authors' Information

Chernyakhovskaya M.Yu. - chernyah@iacp.dvo.ru

Kleschev A.S. - kleschev@iacp.dvo.ru

Moskalenko F.M. - philipmm@yahoo.com

Institute for Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, 5 Radio st., Vladivostok, Russia.

МНОЖЕСТВЕННЫЕ МОДЕЛИ НЕОПРЕДЕЛЁННОСТИ: ЭМПИРИЧЕСКИЙ И МАТЕМАТИЧЕСКИЙ АСПЕКТЫ

Владимир Донченко

Аннотация: *Рассмотрены общие проблемы, связанные неопределённости, включая природу, источники и математические методы её описания - моделирования. Проведена систематизация методов описания неопределённости.*

Ключевые слова*: Неопределённость, обратные задачи, нечёткие множества, преобразование Хока, псевдообращение по Муру – Пенроузу.*

ACM Classification Keywords*: G.3 Probability and statistics, G.1.6. Numerical analysis: Optimization; G.2.m. Discrete mathematics: miscellaneous.*

Conference*: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Вступление

Классическим математическим средством описания неопределённости в её статистическом, «случайном», проявлении является теория вероятностей и математическая статистика (ТВиМС). Вторая половина XX столетия в математике характеризуется интенсивными усилиями по созданию математических средств описания и оперирования с неопределённостью, альтернативных ТВиМС, к которым можно отнести теорию построения оценок с гарантированной точностью (теорию минимаксного оценивания), теорию нечётких множеств, а также преобразование Хока (ПХ). В то же время, многие надежды, связывавшиеся с появившимися теориями, не оправдались. В значительной мере это относится к теории нечётких множеств. Как представляется, многообразие методов математического описания неопределённости, к которым можно отнести: 1) детерминированный, в том числе обратные задачи; 2) статистический; 3) метод получения оценок с гарантированной точностью; 4) метод нечётких множеств; 5) ПХ, – порождает необходимость приведения их к общей основе. Это означает осмысление природы неопределённости, создание общего методологического подхода, который позволил с единой точки зрения рассматривать разнообразные математические методы её описания. В предлагаемой работе предложены некоторые шаги в этом направлении.

Что понимают под наблюдением (экспериментом, опытом, испытанием)?

Понятие неопределенности в поведении исследуемого явления или системы тесно вязано с понятием «опыта», «эксперимента», «наблюдения», «испытания», которые рассматриваются в рамках категории «опыта». Перечисленные понятия имеют общенаучное содержание и часто употребляются как эквивалентные. Кроме того, они употребляются как эквивалентные между собой и в теории вероятностей и математической статистике. В своем общенаучном смысле эти понятия предназначены для описания деятельности, связанной с непосредственной фиксацией фактов на уровне явления: в процессе непосредственного взаимодействия людей с внешним миром

Для выяснения конкретики общенаучного контента определения «эксперимент», «опыт», «наблюдение» обратимся к нейтральному, обезличенному источнику, каковым является, к примеру, БСЭ (Большая советская энциклопедия).

В т. 18 БСЭ на стр.463-464 отмечается, что категория «опыта» совпадает по своей сути с категорией «эксперимента» и «наблюдения». Что касается «эксперимента» и «наблюдения», то в том же издании БСЭ, но в томе 30 на стр.6 в статье, посвященной понятию «эксперимент» отмечается, что термин

происходит от латинского experimentum: проба, опыт, – и означает «метод познания, при помощи которого в контролируемых и управляемых условиях исследуются явления действительности». В той же статье отмечается, что «эксперимент» отличается от «наблюдения» тем, что в первом осуществляется активное оперирование с объектом исследования. Таким образом, в цитируемых источниках понятия «эксперимент» и «наблюдение» различаются активностью или пассивностью в оперировании с исследуемым объектом. В то же время в статье, посвященной «наблюдению» в том же издании БСЭ на стр. 186 в т. 17 отмечается, что наблюдение, обычно, является частью «эксперимента». В той же статье в связи с «наблюдением» появляется сочетание «регистрация наблюдений». Таким образом, можно сделать вывод о том, что опыт или эксперимент является методом познания, который заключается в воссоздании стандартных условий наблюдения исследуемого явления и фиксации соответствующих результатов: того что при созданных условиях появляется. Конечно же, вне поля зрения сознательно оставляется обсуждение вопросов о том, как и каким именно образом обеспечивается создание тех или иных условий, а также вопрос о том, как формируется представление о том, что же именно считать результатом эксперимента.

Эмпирический аспект: наблюдение (эксперимент, опыт, испытание)

Таким образом, из приведённых выше цитирований можно сделать вывод, что принципиальными составляющими «эксперимента» являются:

- воссоздание условий наблюдения для явления, которое исследуется;
- фиксация результатов эксперимента: того, что появляется в результате воспроизведения условий эксперимента..

Воспроизведение условий может носить активный или пассивный характер. В первом случае говорят об «эксперименте», для характеризации второго – употребляют термин «наблюдении». Хотя – подчеркнём ещё раз – оба термина могут употребляться как эквивалентные.

Отметим также, что термин «наблюдение» может употребляться для обозначения части «эксперимента», которая заключается в фиксации (регистрации) результатов «эксперимента».

Анализ статистических подходов к определению эксперимента с небольшими вариациями повторяет выделение приведенных выше основных составляющих эксперименту, выдвигая специфические дополнительные условия в том, что касается «стохастического эксперимента»

«Опыт», «эксперимент», «наблюдение», «испытание»: теоретико-вероятностный контент

Понятия «опыта», «эксперимента», «наблюдения », «испытания» являются также специальными понятиями теории вероятностей. Они имеют, в основном, общий эмпирический контент. И это является естественным, поскольку статистические (теоретико-вероятностные) методы являются признанным математическим средствам моделирования неопределенности в изучении явлений, где она проявляется в виде случайности. Детальнее о случайности, как виде неопределённости, ниже.

В теоретико-вероятностных источниках в определениях понятий «опыт», «эксперимент», «наблюдение» отсутствует единство и единственность в определении понятий. В разных источниках употребляются разные варианты в применении к одним тех же, как можно понять из модельных примеров, объектам. Кроме того, в некоторых источниках обсуждаемые термины употребляются с эпитетами «стохастический», «вероятностный», «случайный». Еще раз заметим, что все они употребляются как эквивалентные. Ниже приводится анализ употребления соответствующих понятий в тех или иных источниках и у тех или иных авторов.

В основополагающей книге А.Н. Колмогорова «Основные понятия теории вероятностей» на стр.12 отмечается, что «применение теории вероятностей к реальному миру опыта происходит в соответствии со следующей схемой.

1. Считается, что имеется определенный комплекс условий $\mathfrak{C}$, который может воспроизводиться неограниченное количество раз.

2. Изучается определенный круг событий, которые могут происходить при воссоздание условий $\mathfrak{C}$.

В таком почтенном источнике, как учебник Б.В. Гнеденко «Курс теории вероятностей», понятие наблюдения в виде «испытания» также связывается с комплексом условий $\mathfrak{C}$. Но – дополнительно – и со связанным с этим комплексом условий набором событий (там же, стр.21). Испытание понимается как воссоздание упомянутого комплекса условий и проверке того, выполняется ли при этом воссоздании условий то или иное событие, выбранное из набора событий (там же, стр. 26). Таким образом «испытания» – это „наблюдение", результаты которого используются для проверки того, выполняется ли исследуемое событие.

У Г. Крамера в классическом издании «Математические методы статистики» на стр. 157-158 понятие „эксперимента" не определяется явно, но выделяется такие, которые могут быть повторенными многократно при одних и тех же условиях. Среди этого типа экспериментов дополнительно выделяются те, для которых в серии экспериментов «результат … может изменяться от одного наблюдения к другому самым неправильным образом». Автор также замечает, что «....в этих случаях мы будем говорить, что имеем дело с последовательностью случайных экспериментов» (там же, стр.158).

В учебнике А.В. Скорохода «Элементы теории вероятностей и случайных процессов» уже в начале книги на стр.5 отмечается, что «одним из основных понятий теории вероятностей является понятие стохастического эксперимента». И дальше указывается, что «так называются эксперименты, результаты которых нельзя предусмотреть». Опять же понятие стохастического эксперимента объясняется примерами (там же, стр.5). А на стр.9 (там же) подчеркивается такая черта стохастических экспериментов, как «возможность повторять их большое число раз». Кроме того, в качестве важной черты стохастического эксперимента, отмечается наличие определенного, не состоящего из одного элемента «числа событий» с ним связанных, среди которых выделяются элементарные события. Отметим, что хотя понятие события не определяется, понятие элементарного события определяется строго (там же, стор.6,7).

В фундаментальном издании, которым является «Справочник по теории вероятностей и математической статистике», изданного авторским коллективом в составе В.С. Королюка, М.И. Портенко, А.В. Скорохода, А.Ф. Турбина, авторы на стр.5 отмечают, что «эксперимент определяется определенным комплексом условий, которые или воспроизводятся искусственно, или осуществляются независимо от воли экспериментатора. Кроме того, они отмечают, что эксперимент определяется также результатами эксперимента, то есть определенными событиями, которые наблюдаются как результат осуществления этого комплекса условий». Авторы также различают «детерминированные» и «случайные» или «вероятностные» эксперименты. К первым относят те, «в которых условия эксперимента однозначно определяют наступление (или не наступление) событий, которые ожидаются». Что же касается «случайных» или «вероятностных экспериментов», то они определяются как такие, в которых «при одних и тех же условиях возможно появление событий, которые исключают друг друга». Отметим, – о чём ниже, – что отмеченных свойств эксперимента недостаточно, чтобы его можно было назвать стохастическим.

В классическом издании «Теория вероятностей» М. Лоева на стр. 13 отмечается, что «...наука имеет дело с закономерностями в испытаниях, которые повторяются», а также что «... долгое время Homo sapiens изучал только детерминированные испытания, в которых условия (причины) полностью определяют результаты (последствия)». Определяются также «случайные испытания» (там же, стр. 13) как такие, в которых при воссоздании их многократно, наблюдаемая частота любого из возможных результатов группируется вокруг определенных чисел. Таким образом, и в этом издании используются понятия, которые связаны с определенным комплексом условий: «испытание» и «случайные испытания», причём последние связываются с теми, в которых частоты появления разных результатов из числа возможных

группируются вокруг определенных чисел. Конечно, обязательной является многоразовость «испытаний и наличие разных, вообще говоря, результатов в разных испытаниях.

В энциклопедическом по широте охвата издании, каковым является двухтомник В.Феллера «Введение в теорию вероятностей и ее приложения», в первом томе, понятие эксперимента считается интуитивно ясным и объясняется большим количеством примеров, которым посвященный §2 первого раздела. Собственно, речь идет о формализованном варианте виртуального опыта в общенаучном понимании, который характеризуется фиксированным набором возможных результатов.

Приведённые выше варианты определения «эксперимента» в теории вероятностей являются типичными и для других изданий, среди которых отметим учебник И.И. Гихмана, А.В.Скорохода, М.Й Ядренко «Теория вероятностей и математическая статистика», монографию «Теория вероятностей. Основные понятия, предельные теоремы, случайные процессы» Ю.В. Прохорова и Ю.А. Розанова, обстоятельный учебник А.Н. Ширяева «Вероятность», учебник И.Н. Коваленко и А.А. Филипповой «Теория вероятностей и математическая статистика», надежный учебник А.А. Боровкова «Курс теории вероятностей». В последнем автор отождествляет случайность с неопределенностью, связывая эту случайность – неопределенность, с незнанием (там же, стр.9), хотя на следующей странице, ссылаясь на принцип неопределенности в физике, отмечает, что неопределенность может быть принципиально свойственной исследуемому явлению. Кроме того, определяя объекты изучения в теории вероятностей (там же, стр.1), автор отмечает, что ими являются явления, для которых, с одной стороны «... те или другие эксперименты или наблюдения могут быть воспроизведены многократно при одинаковых условиях». С другой стороны, на той же странице отмечается, что «теорию вероятностей интересуют те эксперименты, результат которых, выраженный каким-то образом, может изменяться от опыта к опыту». Автор также связывает с результатами определённые события, которые могут в связи с этими результатами рассматриваться, и отмечает, что в этом случае события называют «случайными». Таким образом, в определении „эксперимента" по Боровкову общенаучный контент этого понятия (комплекс условий и, возможность многократного его воссоздания) связывается с изменчивостью результатов от наблюдения к наблюдению. Б.А.Севастьянов в «Курсе теории вероятностей и математической статистики» называет возможность многоразового воссоздания условий в числе ключевых моментов «эксперимента», поскольку именно возможность многократного воссоздания позволяет делать вывод о наступлении или не наступлении тех или иных событий (понятие, которое автором не определяется). Автор называет такие события массовыми, хотя естественнее было бы отнести это название к явлению, по наблюдением которого можно сделать вывод о наступлении или нет исследуемых событий. У У. Гренадера и В. Фрайбергера в «Кратком курсе вычислительной вероятности и статистики» на стр. 10, отмечается возможность повторения эксперимента при одинаковых условиях, а М. Де Гроот в монографии «Оптимальные статистические решения» заявляет, что «статистика как наука занимается теориями и методами, которые используются для принятия решений в условиях неопределенности и неполной информации» (там же, стр.11). А на стр. 14 (там же) отмечается, что «эксперимент употребляется здесь (в работе – *примечание автора*) в самом широком понимании для обозначения, в сущности, любого процесса, все возможные результаты которого могут быть указаны заранее и действительный результат которого является одним из указанных». Автор вводит специальное обозначение S для множества результатов и называет его выборочным пространством эксперимента.

Подводя итоги в определении и использовании понятий «эксперимент», «опыт», «наблюдение», «испытание» – иногда с эпитетами «случайный» или «стохастический» – в теории вероятностей отметим, что они употребляются, как эквивалентные в общем русле общенаучного понимания эксперимента. Специально подчеркиваются и выделяются следующие важные отличительные черты того, что в теории вероятностей и математической статистике понимают под экспериментом.

1. Наличие фиксированного комплекса условий, при котором наблюдается исследуемое явление. Указанный комплекс условий может воспроизводиться активно или пассивно.
2. Возможность многократного воспроизведения комплекса условий (массовость).

3. Возможность описания всех возможных результатов. Это множество результатов называют пространством элементарных событий, выборочным пространством эксперимента и т. п. Вместо множества возможных результатов может рассматриваться множество возможных событий, которые могут иметь или не иметь место в связи с наблюдающимися результатами.
4. Случайность, которая связывается с изменчивостью результатов от эксперимента к эксперименту или с наступлением в разных экспериментах событий, которые исключают друг друга: с непредсказуемостью результатов ли, событий ли от эксперимента к эксперименту.

Основные элементы формализации наблюдения: условия, результат и его регистрация

Как следует из вышеприведенного анализа, двумя основными составляющими эксперимента и в общем эмпирическом контексте и в статистическом понимании являются условия эксперимента (будем называть их также комплексом условий) и его результаты: того, что при этих условиях появляется.

1. Что касается условий эксперимента, то они должны допускать возможность их многоразового воссоздания (массовость). Каждый из возможных комплексов условий, при котором можно проводить эксперимент будем обозначать κ, соответственно – через K будем обозначать совокупность разных вариантов возможных комплексов условий, при которых можно проводить эксперимент.
2. Результат эксперимента, будем обозначать его y, – это то, что может появиться при воспроизведении комплекса условий $\kappa \in K$. Через Y_κ будем обозначать, вообще говоря, – множество всех возможных результатов, которые могут появиться при воспроизведении условий κ, поскольку фиксация комплекса условий, вообще говоря, не гарантирует однозначного результата эксперимента.

Важно отметить, что термин «результат эксперимента» для обозначения того, что может появиться при воспроизведении комплекса условий (то, что выше обозначается через y), часто употребляют с другим контентом: в смысле фиксации результатов экспериментов, что, как отмечалось выше, иногда понимают как часть эксперимента и обозначают также термином «наблюдение», понимаемом в узком смысле.

В дальнейшем «регистрация результатов» будет рассматриваться как составляющая эксперимента.

Определение 1. Регистрацией результата эксперимента будем назвать фиксацию того, что определяет две составляющие эксперимента: условия κ и результат y – т.е. пару $s = (\kappa, y)$ «условие-результат». Соответственно, под регистрацией серии из N экспериментов (выборкой) будет пониматься последовательность пар

$$s_1,...,s_N = (\kappa_1, y_1),...,(\kappa_N, y_N). \quad (1)$$

Замечание 1. Как показывает анализ, результат эксперимента, как значение y, часто не отличают от «регистрации результата эксперимента» как того, что фиксируется в связи с проведённым экспериментом и обозначают одним и тем же термином «результат эксперимента».

Основные составляющие наблюдения: детализация условий

Необходимость учёта в одной серии экспериментов с разными условиями привела к необходимости структуризации условий наблюдения. Такая структуризация обеспечивает контролируемое изменение условий наблюдения от одного эксперимента серии к другому. При таком изменении часть комплекса условий остается, вообще говоря, неизменной по умолчанию, а часть изменяется контролируемым образом. Собственно: это означает, что любой из возможных комплексов $\kappa \in K$ условий проведения эксперимента представляется парой $\kappa = (x, f)$, в которой $x \in X$ обозначает вариативную, изменяемую от эксперимента к эксперименту часть, а f – неизменную по умолчанию для серии экспериментов часть комплексов условий наблюдения.

Определение 2. Экспериментом с управляемыми условиями (УпрУЭкс) будем называть такой, в котором условия представляются в виде $\kappa = (x, f)$, $x \in X, f \in \Im$, $K = X \times \Im$, x будем называть вариативной частью условий, f - частью условий по умолчанию, κ - полными условиями эксперимента.

Замечание 2. Отметим, что при детерминированном подходе неизменная часть условий ассоциируется с однозначностью связи результата наблюдения с вариативной частью условий, т.е. – с функцией (функцией отклика) от вариативной, изменяемой части условий. Само же исследование в рамках УпрУЭкс экспериментов в литературе называют моделью «вход-выход» системы с очевидным делением на то, что называют входом, выходом и функцией отклика системы.

Регистрация наблюдений: практика

Следует отметить, в экспериментальной практике регистрация эксперимента в смысле определения 1 подменяется другими вариантами того, что называют фиксацией $s_1,...s_N$. Такими вариантами в рамках выделенных выше составляющих эксперимента могут быть следующие:

$$s_1,...,s_N = \begin{cases} y_1,...,y_N \\ (x_1,y_1),...,(x_N,y_N) \\ (\kappa_1,y_1),...,(\kappa_N,y_N) \end{cases}. \quad (2)$$

Множественные модели неопределённости (МнМоН)

Определение 3. В рамках введённых выше понятий множественными моделями неопределённости для исследуемого явления будем называть такое описание неопределённости, которое базируется на множественности значений Y, Y_x, Y_κ: того, что появляется или может появиться в результате серии экспериментов в (2). В зависимости от того, как понимается регистрация эксперимента, это может быть:

$$Y = \bigcup_{i=1}^{N} \{y_i\},$$
$$Y_x = \bigcup_{i:x_i=x} \{y_i\}, x \in \bigcup_{i=1}^{N} \{x_i\}, \quad (3)$$
$$Y_\kappa = \bigcup_{i:\kappa_i=x} \{y_i\}, \kappa \in \bigcup_{i=1}^{N} \{\kappa_i\}$$

Собственно, (3) фиксирует множественность значений того, что может появиться при фиксированном комплексе условий. В первом случае из (3) комплекс условий по умолчанию является одинаковым для всех экспериментов серии, во втором – в серии экспериментов условия варьируются. Множество возможных вариантов условий в серии экспериментов определяется множеством $\bigcup_{i=1}^{N} \{x_i\}$, а Y_x определяет множество значений *y* в тех экспериментах серии, в которой вариативная часть одна и та же и определяется вариативной частью $x \in \bigcup_{i=1}^{N} \{x_i\}$. В третьем варианте то же касается $\bigcup_{i=1}^{N} \{\kappa_i\}$, которое описывает множества всех возможных вариантов условий серии.

МнМоН: неопределённость в детерминированных наблюдениях

Для детерминированности характерна однозначная связь «полные условия – результат»:

$$\kappa \to y_\kappa,\ Y_\kappa = \{y_\kappa\}, \kappa \in K. \quad (4)$$

Однако, если в серии экспериментов условия изменчивы, а регистрация проводится в виде $y_1,...,y_N$ вместо $(x_1,y_1),...,(x_N,y_N)$ или $(\kappa_1,y_1),...,(\kappa_N,y_N)$, то возникает множественность (не одноэлементность) Y, которая, собственно, и является неопределённостью. Такая неопределённость в детерминированном эксперименте связана с проблемой скрытых параметров: дополнительных условий, которые нужно учитывать при регистрации эксперимента, чтобы явление стало детерминированным, т.е., чтобы наблюдения могли быть охарактеризованными в соответствии с (4).

МнМоН: случайность

Случайность в исследуемом явлении с одной стороны характеризуется тем, что связь возможных результатов с полными условиями эксперимента в (3) неоднозначна: для каждого из фиксированных условий в разных экспериментах могут появляться разные результаты:

$$\kappa \to Y_\kappa \neq \text{"одноэлементное множество"},\ \kappa \in \mathrm{K}.$$

С другой стороны, для явления, которое называют случайным (а сам эксперимент – стохастическим), должен выполняться закон устойчивости частот. Этим термином обозначается предположение о том, что частоты тех или иных групп возможных результатов должны сходиться к предельному значению, которое на должно зависеть от серии экспериментов, по которому оно получено, но характеризовать само исследуемое явление: быть одинаковым для разных серий экспериментов.

МнМоН: гарантированные оценки (минимакс)

Этот подход связан с дальнейшей априорной структуризацией вариативной или функциональной части условий в рамках детерминированного описания явления: $x=(x^{(1)},x_V^{(2)})$ или $f=(f^{(1)},f^{(2)})$ и предположением о том, что в эксперименте фиксируется только одна из частей, например $x^{(1)}$ или $f^{(1)}$ (наблюдаемая компонента), а про вторую – известно, что она принадлежит множеству $E_{x^{(1)}}$ или $E_{f^{(1)}}$ соответственно, которое определяется наблюдаемой компонентой.

МнМоН: интервальный подход

В модель наблюдений со структурированной вариативной частью очевидным образом вкладывается интервальная модель неопределённости. Действительно, достаточно предположить, что в обозначениях предыдущего пункта

$$y = f(x^{(1)}) + x^{(2)},\ x^{(2)} \in (-\Delta_{x^{(1)}},\Delta_{x^{(1)}}) = E_{x^{(1)}}.$$

МнМоН: нечёткие множества

Место нечёткости [Zadeh, 1962] во множественных моделях неопределённости может быть определено в рамках статистической интерпретации нечетких множеств [Donchenko, 1998, a) b)]. Эта интерпретация определяется следующей теоремой.

Теорема [Donchenko, 1998 a),b)]. Для нечёткого множества, задаваемого парой (E,μ) носитель-функция принадлежности, в случае, когда E - пространство с мерой, а μ- измерима, можно построить вероятностное пространство (Ω,B_Ω,P), событие $A\in B_\Omega$, полную группу событий $H_e=\{\eta=e\},e\in E$, $\eta - E$ - значная случайная величина, так, что

$\mu(e)=P(A\,|\,H_e),e\in E$.

МнМоН: обратные задачи

Важным классом неопределённостей в детерминированных задачах являются обратные задачи, т.е. задачи в которых необходимо определить множество возможных вариативных частей условий (входов), которые обеспечивают заданное значение результата (выхода). Отметим важную роль псевдообращения по Муру - Пенроузу [Алберт, 1977] и его развитию в работе [Кириченко, 1997] для линейных задач и для применения в задачах кластеризации и распознавания образов [Кириченко, Донченко., 2007].

МнМоН: преобразование Хока

Специальным случаем неопределённости является ПХ [Hough, 1962]. Этот вид неопределенности порождён множественностью возможных вариантов функций отклика в наблюдениях: $\kappa_i = (x_i, f_i), i = \overline{1,N}$. Простейшей моделью наблюдений такого рода может служить бинаризованное изображение, на котором представлено несколько прямых. Выборка представляет собой координаты точек изображения с единичным значением яркости. Детальнее с ПХ и его математической формализацией можно познакомиться в [Donchenko, 2003].

Заключение

Предложенная в работе концепция «множественных моделей неопределённости» связывает неопределённость с экспериментом и позволяет на единой основе рассматривать многообразие математических методов описания неопределённости. В работе также определено место каждого из математических методов в рамках предложенной концепции.

Литература

[Алберт, 1977] Алберт А. Регрессия, псевдоинверсия, рекуррентное оценивание. – М.: Наука. – 1977.– 305 с.

[Donchenko, 2003] Donchenko V.S. Hough Transform and Uncertainty// Proceedings X International Conference “Knowledge – Dialog – Solution”. – June 16-23, 2003. – Varna (Bulgaria). – P.391-395.

[Донченко,1968,a)] Донченко В.С. Умовні розподіли та нечіткі множини // Вісник Київського університету. – 1998.–Вип. №3 – С. 175-179.

[Донченко,1968,b) Донченко В.С. Імовірність та нечіткі множини .// Вісник Київського університету. Серія фізико-математичні науки. – 1998. – Вип. №4. – С. 141-144

[Hough] Hough P.V.C. Method and Means for Recognizing Complex Patterns. - U.S. Patent 3069354. – December 1962.

[Кириченко, 1997] Кириченко Н.Ф. Аналитическое представление псевдообратных матриц //Киб. и СА.- №2. –1997.– С.98-122.

[Кириченко, Донченко, 2007] Кириченко Н.Ф., Донченко. В.С. Псевдообращение в задачах кластеризации// Киб. и СА.- №4, 2007– С.98-122.

Zadeh, Lotfi. Fuzzy Sets// Information and Control. – June, 1965. – 8(3).–P. 338-353.

Информация об авторе

Владимир С. Донченко – *Профессор; Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, Украина, e-mail: voldon@unicyb.kiev.ua*

ОПИСАНИЕ ФИЗИЧЕСКИХ ЯВЛЕНИЙ ГИПЕРСЛУЧАЙНЫМИ МОДЕЛЯМИ

Игорь Горбань

***Аннотация**: Проведен анализ публикаций, посвященных теории гиперслучайных событий, величин, процессов и полей. Показано, что рассматриваемая теория относится к классу теорий, описывающих математические модели, построенные конструктивным образом. От нее нельзя ожидать получение новых решений, не эквивалентных решениям, следующим из теории вероятностей и математической статистики. Однако, подобно теории матриц, она расширяет возможности решения практических задач. Гиперслучайные модели, учитывающие возможности изменения законов распределения событий, величин, процессов и полей, более адекватно описывают реальные ситуации, чем случайные модели с фиксированными законами распределения. Установлено, что следствием выдвинутой ранее гипотезы о том, что все реальные явления (за исключением возможно лишь мировых физических констант) носят гиперслучайный характер, является то, что абсолютно все оценки реальных величин, функций и полей не состоятельны и потенциальная точность любых измерений ограничена. Кроме того, абсолютно достоверное обнаружение и абсолютно достоверная классификация реальных объектов принципиально невозможны.*

***Ключевые слова**: гиперслучайная модель, гиперслучайная величина, точность измерения, состоятельность.*

***ACM Classification Keywords**: G.3 Probability and Statistics*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Каждый человек воспринимает окружающий мир по-своему в соответствии с своим жизненным опытом, полом, профессией, возрастом и пр. Рассказывая или анализируя что-либо, мы обращаем внимание на то, что представляется нам наиболее существенным. Это – первый шаг формализации реальных явлений, завершающийся формированием физических моделей с использованием разнообразных физических событий, величин, процессов и полей. Следующим шагом формализации является математическое описание этих физических моделей.

Одной из основных проблем познания является адекватное описание реального мира. Физические и математические модели, используемые для описания различных явлений, постоянно совершенствуются. С получением новых данных и развитием представлений о мире старые модели отходят на второй план, заменяются новыми, более совершенными.

Если до конца средневековья мир виделся неизменным и описывался преимущественно детерминированными моделями, то в настоящее время он рассматривается как динамично меняющаяся структура. Главным средством математического описания мира стали модели, использующие в качестве абстрактных математических объектов случайные явления – случайные события, величины, процессы и поля.

В данном случае термин «случайный» используется в математическом смысле [1, 2]. Наиболее полно случайное явлений можно охарактеризовать с помощью функции распределения, определяемой для строго фиксированных статистических условий наблюдения. Наличие определенной вероятности является характерной чертой любого случайной величины и, вообще, любого случайного явления.

Если на интервале наблюдения явления (который может быть временным, пространственным, пространственно-временным или иным) статистические условия практически не меняются, то возможно

построение корректной стохастической модели с использованием случайных величин и функций с некоторыми, хотя, возможно, и не всегда известными законами распределения.

На практике такая возможность существует не очень часто. Реальные физические явления находятся в неразрывной связи со всем окружающим миром. Изменения в мире приводят к изменениям статистических условий наблюдения, причем к изменениям, носящих непредсказуемый характер. Обеспечить на практике полную стабильность условий нельзя, а, следовательно, нельзя задать или оценить абсолютно точно функцию распределения. Это накладывает ограничения на применение классических случайных моделей с фиксированным законом распределения. Корректное их использование оказывается возможным лишь при пренебрежимо малых изменениях условий наблюдения.

Вариабельность условий наблюдения – серьезная проблема, мешающая построению корректных математических моделей. Не очень обоснованный, но очень распространенный прием решения проблемы – игнорирование факта изменения условий. При этом статистически неопределенную или статистически нестабильную величину или процесс представляют стохастической моделью, которая описывается определенным законом распределения.

Такой паллиатив иногда оказывается приемлемым, однако, не всегда. Для комплексного решения проблемы необходимы подходы, позволяющие описывать явления не только в определенных и статистически стабильных условиях. Многочисленные непараметрические методы обработки (ранговые, знаковые, робастные и др.) [3 – 10] ориентированы именно на такой подход. С их помощью удается выделить те особенности физических явлений, которые не связаны или слабо связаны с изменением статистических условий наблюдения. Известным недостатком непараметрических методов является эмпирический их характер [6].

Стремление найти простые средства учета неопределенности условий наблюдения играла, по всей видимости, не последнюю роль при формировании ряда относительно новых научных направлений, таких как теория нечетких множеств [11], теория нейронных сетей [12], теория хаотических динамических систем [13, 14], теория интервальных данных [15, 16] и др.

Постоянно продолжающийся поиск универсальных и эффективных путей адекватного математического описания явлений недавно привел к новому классу моделей, в которых в качестве абстрактных математических объектов выступают, так называемые, гиперслучайные явления [17].

Под гиперслучайными явлениями подразумеваются семейство случайных событий, величин, функций или полей, зависящие от параметра $g \in G$, который рассматривается как независимая переменная и ассоциируется с условиями наблюдения (или условиями формирования) рассматриваемых объектов.

Математически случайные события описываются с помощью вероятностного пространства [1], задаваемого триадой $(\Omega, \Im, P)$, где Ω – пространство элементарных событий $\omega \in \Omega$, $\Im$ – борелевское поле (σ – алгебра подмножеств событий) и P – вероятностная мера подмножеств событий.

При менее строгом, но более наглядном статистическом определении (по Р. фон Мизесу [18, 19]), вероятность $P(A)$ случайного события A представляется как предел частоты $p_N(A)$ его появления при проведении опытов в одинаковых условиях и устремлении количества опытов N к бесконечности: $P(A) = \lim_{N \to \infty} p_N(A)$. При небольших значениях N частота $p_N(A)$ может колебаться, однако по мере увеличения N постепенно стабилизируется и при $N \to \infty$ стремится к определенному пределу $P(A)$.

Гиперслучайные явления можно описать с помощью тетрады $(\Omega, \Im, G, P_g)$ [17], где Ω – пространство элементарных событий $\omega \in \Omega$, $\Im$ – борелевское поле, G – множество условий $g \in G$, P_g – вероятностная мера подмножеств событий, зависящая от условия g. Таким образом, вероятностная

мера задается для всех подмножеств событий и всех возможных условий $g \in G$. Мера же для условий $g \in G$ остается не определенной.

Используя менее строгий статистический поход, гиперслучайное событие A можно трактовать как событие, частота появления которого $p_N(A)$ при увеличении числа опытов N не стабилизируется и при $N \to \infty$ не имеет предела.

Разработке основ теории гиперслучайных явлений посвящен цикл статей автора, в частности [17, 20 – 24], и монография [25]. Главное внимание в этих работах уделялось математическим аспектам новой теории; вопросы же представления реальных физических явлений гиперслучайными моделями оставались несколько в тени.

Целью настоящей работы является краткий обзор опубликованных материалов и исследование трех ключевых, на наш взгляд, вопросов, а именно: 1) какие новые возможности предоставляет теория гиперслучайных явлений и чем они обусловлены, 2) какие модели – случайные или гиперслучайные – более адекватно описывают реальные явления и 3) какие практические результаты следуют из новой теории?

Обзор результатов по теории гиперслучайных явлений

В первой статье, касающейся теории гиперслучайных явлений [17], введены математические понятия гиперслучайного события и гиперслучайной величины. Для характеристики гиперслучайного события предложено использовать верхнюю и нижнюю границы вероятности, а для описания гиперслучайной величины – верхнюю и нижнюю границы функции распределения. Определены понятия плотности распределения границ гиперслучайной величины, характеристических функций границ, а также нецентральных и центральных моментов границ. Исследованы свойства этих характеристик. Результаты, первоначально полученные для скалярных действительных гиперслучайных величин, обобщены на случай комплексных и векторных величин. Работа [20] посвящена гиперслучайным функциям. Для их описания разработан математический аппарат, базирующийся на принципах и подходах описания гиперслучайных величин. В качестве основных характеристик гиперслучайных функций выбраны верхняя и нижняя границы функции распределения, а также плотности распределения границ и характеристические функции границ. Вспомогательными характеристиками являются математические ожидания границ, дисперсии границ, корреляционные и ковариационные функции границ и др. Математический аппарат разработан для различных гиперслучайных функций: скалярных и векторных, вещественных и комплексных. Изучена специфика описания скалярных, векторных и комплексных гиперслучайных функций.

В статье [21] рассмотрен альтернативный подход к описанию гиперслучайных величин и функций, позволяющий характеризовать их, не прибегая к сложному расчету границ функции распределения. Его основой служит вычисление границ центральных и нецентральных моментов: математического ожидания, дисперсии, корреляционного момента, ковариационного момента и др. Границы моментов гиперслучайного явления и моменты границ функции распределения – разные понятия. В общем случае границы моментов не совпадают с соответствующими моментами границ, хотя в отдельных случаях совпадение и имеет место. Работа [22] посвящена формализации ряда понятий, касающихся гиперслучайных явлений: стационарности и эргодичности гиперслучайной функции, гиперслучайного белого шума и др. Введены различные характеристики, описывающие стационарные и эргодические гиперслучайные функции. Исследованы свойства этих характеристик. Разработаны спектральные методы описания стационарных гиперслучайных функций.

В статье [23] формализовано понятие гиперслучайной выборки и определены ее свойства. Предложена методология формирования оценок характеристик гиперслучайной величины для случая, когда условия меняются достаточно медленно и существует некий интервал, в течение которого условия можно считать практически неизменными. Тогда данные, полученные на этом интервале, можно ассоциировать с

определенными, хотя и неизвестными, условиями. Исследована сходимость гиперслучайных оценок к соответствующим точным характеристикам. Для гиперслучайных оценок найдены условия сходимости оценок по вероятности к точным значениям. Доказана предельная теорема, определяющая закон распределения оценок границ функции распределения среднего при стремлении объема выборки к бесконечности.

Работа [24] посвящена методам точечного и интервального оценивания параметров гиперслучайных величин. Для точечных гиперслучайных оценок введены понятия несмещенной оценки, состоятельной, эффективной и достаточной, а для интервальных гиперслучайных оценок – понятия доверительного интервала и границ доверительной вероятности. Доказаны теоремы, определяющие границы нижней границы точности точечной оценки и границы доверительного интервала интервальной оценки.

В монографии [25] обобщены полученные научные результаты.

Конструктивный характер теории гиперслучайных явлений

Реальные объекты и их связи описываются с помощью физических моделей, которые могут быть представлены различными математическими моделями, содержащими разные математические объекты и отношения между ними.

Любая математическая модель со своими математическими объектами и отношениями (в частном случае операциями) определяется набором непротиворечивых и независимых аксиом. Новая математическая модель может быть построена либо путем введения новых объектов, отношений и аксиом либо конструктивно с использованием уже известных математических понятий.

Примерами теорий, соответствующих конструктивным моделям, могут служить векторная алгебра или теория матриц, основанные на правилах сложения и умножения чисел.

Теория гиперслучайных явлений представляет собой теорию, описывающую математическую модель, построенную на базе математической модели случайных явлений. Все новые объекты и отношения между ними строятся с помощью известных объектов и отношений. Никакие новые аксиомы не вводятся. Поэтому теорию гиперслучайных явлений следует рассматривать как теорию, соответствующую конструктивно построенной математической модели.

Следует отметить интересный факт, что решение любой задачи, описываемой средствами конструктивной математической модели, эквивалентно решению этой же задачи, описываемой средствами порождающей модели. Решение, получаемое, например, с помощью теории матриц, полностью эквивалентно решению, использующему лишь арифметические правила работы с числами.

В этой связи от теории гиперслучайных явлений нельзя ожидать получение новых решений, не эквивалентных решениям, получаемых с использованием методов теории вероятностей и математической статистики.

Вместе с тем, не следует умалять значимость новой теории. Несмотря на указанные ограничения, накладываемые конструктивным характером модели, теория гиперслучайных явлений, подобно теории матриц, расширяет возможности решения практических задач. Использование в новой теории обобщенных понятий позволяет взглянуть на существующие проблемы с более высоких позиций и уловить те закономерности и особенности исследуемых явлений, которые на уровне понятий порождающей модели были скрыты громоздкостью рассуждений и выкладок. Кроме того, процедура решения некоторых задач оказывается более простой и наглядной, а само решение представимо в более компактном виде.

Случайные и гиперслучайные модели реальных явлений

Как правило, реальные явления более адекватно представляются гиперслучайными моделями, чем случайными. В подтверждение этому разберем классический пример, с которого обычно начинается

изучение теории вероятностей, – пример с подбрасыванием монеты. Обычно полагают, что исходы опытов – случайны и имеют конкретные вероятности: вероятность того, что выпадет орел, – P_o, а вероятность того, что выпадет решка, – P_p. При этом считают, что $P_o = P_p = 0,5$.

Насколько корректна такая модель? На первый взгляд, кажется, что нет оснований сомневаться в ее адекватности. Но это только на первый взгляд. Разве вероятности P_o, P_p не могут быть другими? Нетрудно убедиться, что после некоторой тренировки, контролируя начальное положение монеты, можно научиться так ее бросать, что частота выпадения одной из ее сторон будет колебаться в районе определенного фиксированного значения, большего 0,5, а частота выпадения второй стороны – в районе значения, меньшего 0,5. При изменении условий бросания показатели могут меняться, как в одну, так и другую сторону. Исходы опытов в этом случае можно рассматривать как гиперслучайное событие. Таким образом, гиперслучайная модель, учитывающая возможности изменения вероятностей выпадения орла и решки, более адекватно описывает реальную ситуацию, чем случайная модель, предполагающая фиксированные значения этих вероятностей.

Рассмотрим другой пример, имеющий отношение к метрологии, – прецизионное измерение диаметра цилиндрической детали круглого сечения. Совершенно тривиальная задача при углубленном анализе оказывается очень непростой. Изготовить деталь абсолютно круглого сечения невозможно. Ее сечение всегда отличается от идеального круга: во-первых, из-за эллипсоидального или иного отклонения от идеальной круговой формы, а, во-вторых, из-за шершавости поверхности. Следует также иметь в виду, что разные сечения по оси цилиндра отличаются. Поэтому истинный размер детали, даже без учета влияния температуры и целого ряда других факторов, которые в дальнейшем в интересах упрощения рассуждений будем игнорировать, может быть разным в разных замерах.

Из-за сложной формы сечения детали понятие диаметра в данном случае оказывается не приемлемым. Принимая во внимание это обстоятельство, задачу следует формулировать как задачу измерения размера сечения.

Физическая модель измеряемой величины должна учитывать и отклонение от идеальной круговой формы, и шершавость, и различие сечений по оси. Для математического описания физической модели можно использовать как общепринятую случайную, так и гиперслучайную математическую модель.

Случайная модель базируется на предположении, что вероятностные характеристики результатов измерения постоянны. В действительности же это не так. В пределах небольших локальных областей они могут быть приблизительно постоянными, однако в целом могут существенно зависеть от направления, вдоль которого проводится измерение, и рассматриваемого сечения. Поэтому гиперслучайная модель, учитывающая вариабельность функций распределения, лучше описывает измеряемую величину, чем случайная модель.

Любые измерения проводятся в условиях воздействия различных мешающих факторов (помех). В рассматриваемой задаче в качестве таковых выступает загрязнение поверхности детали. Пыль и грязь на поверхности собирается неравномерно. В пределах небольших локальных областей загрязнение носит случайный характер, однако, в целом, из-за отличия законов распределения для разных областей – гиперслучайный характер.

Идеальных, абсолютно точных, средств измерения в мире не существует. Ни штангенциркуль, ни микрометр, ни какой-то другой измерительный инструмент не может с бесконечно высокой точностью измерить размер сечения. Причины разные – шероховатость поверхности детали, наличие разнообразных инструментальных ошибок и пр., но объединяющим свойством оказывается гиперслучайный характер этих причин. Поэтому, строя математическую модель средства измерения, и здесь имеет смысл отдавать предпочтение гиперслучайной модели.

Отсюда следует, что обобщенная физическая модель, учитывающая в комплексе реальные особенности измеряемой величины, помехи и средства измерения, описывается более адекватно гиперслучайной математической моделью, чем случайной.

Применение теории гиперслучайных явлений

Описанные выше ситуации типичны для очень широкого круга реальных явлений, что позволяет предположить [25], что все реальные явления (за исключением, возможно лишь, небольшое число величин, рассматриваемых современной наукой как мировые физические константы) носят гиперслучайный характер. Иными словами, одним из основных физических свойств практически всех реальных событий, величин, процессов и полей является неполная их статистическая определенность.

Выдвинутая гипотеза касается различных реальных явлений, в том числе величин, функций и полей, подлежащих измерению, действующих помех, а также инструментальных ошибок измерения. Любая оценка формируется в результате оценивания средствами измерения смеси истинного значения измеряемой величины, функции или поля и помех, мешающих проведению наблюдений. Поскольку помеха и инструментальные ошибки измерения носят гиперслучайный характер, даже в том случае, когда измеряемая величина постоянна (является мировой константой), результат измерения оказывается величиной гиперслучайного типа. Отсюда следует [25], что абсолютно все оценки реальных величин, процессов и полей не состоятельны и потенциальная точность любых измерений ограничена. Предел точности определяется не только числом результатов измерения и случайным их разбросом, а и изменчивым характером вероятностных характеристик измеряемой величины, действующих помех и инструментальных ошибок.

Задачи обнаружения и классификации могут рассматриваться как задача измерение некоторых параметров. Поэтому абсолютно достоверное обнаружение и абсолютно достоверная классификация реальных объектов принципиально невозможны.

Выводы

1. Теория гиперслучайных явлений достаточно быстро прошла первоначальный этап становления и в настоящее время претендует на роль математической теории, ориентированной на адекватное описание реальных физических явлений.

2. Теория гиперслучайных явлений относится к классу теорий, описывающих модели конструктивного типа. Поэтому от нее нельзя ожидать получение новых решений, не эквивалентных решениям, следующим из теории, соответствующей порождающей ее модели, в данном случае теории вероятностей и математической статистики.

3. Теория гиперслучайных явлений, подобно теории матриц, расширяет горизонты для решения практических задач. Использование в новой теории обобщенных понятий позволяет взглянуть на существующие проблемы с более высоких позиций и уловить закономерности и особенности исследуемых явлений, которые на уровне понятий порождающей модели были скрыты громоздкостью рассуждений и выкладок. Кроме того, процедура решения некоторых задач оказывается более простой и наглядной, а само решение представимо в более компактном виде.

4. Гиперслучайные модели, учитывающие возможности изменения законов распределения событий, величин, процессов и полей, более адекватно описывает реальную ситуацию, чем случайные модели, предполагающие фиксированные значения этих законов.

5. Проведенные исследования показывают, что практически все реальные явления (за исключением возможно лишь мировых физических констант) носят гиперслучайный характер, абсолютно все оценки реальных величин, процессов и полей не состоятельны и потенциальная точность любых измерений ограничена. Кроме того, абсолютно достоверное обнаружение и абсолютно достоверная классификация реальных объектов принципиально невозможны.

Литература

1. Колмогоров А.Н. Основные понятия теории вероятностей. ОНТИ, 1936.
2. International standard ISO 3534-2: 2006 (E/F). Statistics – Vocabulary and symbols – Part 2: Applied statistics. P. 125.
3. Леман Е. Проверка статистических гипотез. Пер. с англ./Пер. Ю.В. Прохорова. М.: Наука, 1971. 375 с.
4. Королюк В.С. и др. Справочник по теории вероятностей и математическая статистика. М.: Наука, 1985. – 637 с.
5. Левин Б.Р. Теоретические основы статистической радиотехники. Т. 3. М., Сов. радио, 1976, 285 с.
6. Под ред. Бакута П. А. Теория обнаружения сигналов. М., «Радио и связь», 1984. С.440.
7. Ван Трис Г. Теория обнаружения, оценок и модуляции. – М.: Сов. радио, 1972. Т.1. 743 с.; 1975. Т.2. 343с.; 1977. Т.3. 662с.
8. Хьюбер П. Робастность в статистике. М., Мир, 1984. 303с.
9. Кнопов П.С., Голодников А.Н., Пепеляев В.А. Оценивание параметров надежности при наличии неполной первичной информации – Компьютерная математика, №1, 2003. С. 36 –47.
10. Кравцов Ю.А. Случайность, детерминированность, предсказуемость. Успехи физических наук, т. 158, вып. 1, 1989. С. 93–122.
11. L.A. Zadeh and J. Kacprzyk (Eds.), “Fuzzy logic for the management of uncertainty,” John Wiley & Sons, New York, 1992. 256c.
12. M.T. Hagan, H.B. Demuth, and M.H. Beale, “Neural network design,” Boston, MA: PWS Publishing, 1996. 252 c.
13. R.M. Crownover, “Introduction to fractals and chaos,” Jones and Bartlett Pub., Inc., Boston – London, 1995. 195 c.
14. Гринченко В.Т., Мацыпура В.Т., Снарский А.А.. Введение в нелинейную динамику. К.: Наукова думка, 2005. 263с.
15. Орлов А.И. Эконометрика. Учебник. М.: «Экзамен».– 2002. – 576с.
16. Левин В.И. Интервальная математика и изучение неопределенных систем// Информационные технологии. – 1998. №6. (Федеральный портал «Инженерное образование». Интеллектуальные системы. 5 мая 2005. www.techno.edu.ru).
17. Горбань И.И. Гиперслучайные явления и их описание //Акустичний вісник. 2005. т.8, № 1–2. С.16–27.
18. R. von Mises, “Mathematical theory of probability and statistics,” Edited and complemented by H. Geiringer. N.Y. and London, Acad. Press, 1964. 232c.
19. Гнеденко Б.В. Курс теории вероятностей. М.: И-во физмат. литературы. 1961. 406 с.
20. Горбань И.И. Гиперслучайные функции и их описание //Радиоэлектроника. 2006. № 1. С.3–15.
21. Горбань И.И. Методы описания гиперслучайных величин и функций. // Акустичний вісник. 2005. т. 8, № 3. С.24-33.
22. Горбань И.И. Стационарные и эргодические гиперслучайные функции. //Радиоэлектроника. 2006. № 6. С.54-70.
23. Горбань И.И. Оценки характеристик гиперслучайных величин //Математические машины и системы. 2006. № 1. С.40-48.
24. Горбань И.И. Точечный и интервальный методы оценки параметров гиперслучайных величин. //Математические машины и системы. 2006. № 2. С.3 –14.
25. Горбань И.И. Теория гиперслучайных явлений. К.: ИПММС НАН Украины. 2007. 184 с.

Сведения об авторе

Горбань Игорь Ильич *– заместитель генерального директора по научной работе ГП „УкрНИУЦ”, доктор технических наук, профессор, Украина, Киев, ул. Святошинская, 2;.*
e-mail: gorban@ukrndnc.org.ua.

ИНФОРМАЦИЯ И МОДЕЛИ[1]

Виктор Неделько

***Аннотация**: В работе обсуждается возможность формализации базовых понятий интеллектуальной деятельности, таких как информация, смысл (семантика), интеллект. Основная трудность этой задачи в том, чтобы достичь достаточно строгой математической формализации, сохранив содержательный смысл исходных понятий. Для достижения данной цели в работе используется подход, основанный на теории моделей, который позволяет формализовать данные понятия, определив их в конечном счете как некоторые множества. Исследуется возможность использования теории моделей в задаче понимания естественного языка и для оценивания семантической близости высказываний.*

***Ключевые слова**: информация, теория моделей, искусственный интеллект, экспертные системы, обработка естественного языка, семантика, высказывания на естественном языке.*

***ACM Classification Keywords**: I.2.0 Artificial intelligence – Philosophical foundations, I.2.7 Natural Language Processing, H.1 Information Systems – Models and principles.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Целью данной работы является исследование возможности формализации основных понятий, связанных с интеллектуальной деятельностью, в частности, понятий информации, смысла высказывания, естественного и искусственного интеллекта.

Подобные формализации, вообще говоря, достаточно давно известны, например, теория информации, машины Тьюринга, однако они несут, как правило, существенно более узкий смысл, чем тот, который вкладывается в соответствующие понятия вне этих теорий. В данной работе представлена попытка избежать обеднения таких понятий при формализации, возможно, за счет отступления от математической строгости, но добиться, чтобы определения имели математический смысл, а именно, так или иначе определяли некоторое множество объектов с заданными свойствами.

Работа в значительной мере носит философский характер, но, пожалуй, в меньшей, чем может показаться на первый взгляд. Так многие моменты, которые могут показаться изложением идеалистических философских концепций, являются, как правило, не более чем частичной математической формализацией предметной области.

Базовые понятия

Базовыми понятиями для описания интеллектуальной деятельности являются понятия субъекта и объекта, которые мы можем в контексте данной работы отождествить с первичными философскими категориями «Я» и «не Я». «Я» – есть в первую очередь сознание, которое можно представить как совокупность ощущений (образов). Ощущения здесь – не только сигналы от органов чувств, но и мысленные образы (фактически такие же сигналы, только сымитированные мозгом).

Рассмотрим множество субъектов. Мироощущение каждого индивидуально, но для них имеет место некоторая эквивалентность.

Предположим, два субъекта сидят за столом и смотрят на книгу в зеленом переплете. Возможно, что их восприятие существенно различается, например, может оказаться, что зеленый цвет одним воспринимается так же, как другим красный. Однако установить это различие можно только, если произвести обмен сознаниями. Поскольку замена сознания возможна лишь умозрительно, для этих лиц

[1] Работа выполнена при поддержке РФФИ, грант № 07-01-00331-а.

возможно договориться, что они оба наблюдают зеленую книгу, и это ни в чем не приведет к противоречиям.

Таким образом, есть имеет место «изоморфизм» восприятия. Все множество ощущений можно факторизовать по отношению эквивалентности, иными словами, сопоставить каждому классу эквивалентных ощущений (образов) элемент некоторого множества. Полученное фактор-пространство назовем множеством моделей или универсумом.

Часть универсума, которая является общей для большинства субъектов, назовем реальностью.

Заметим, что такое определение реальности вовсе не подразумевает идеалистического подхода и не зависит от философских концепций. Философские аспекты вопроса более подробно обсуждаются в последнем разделе.

Связь с теорией моделей

Если говорить неформально, семантика высказывания — это те мысли, которые хотел передать автор. Однако мысли — сущность, недоступная для наблюдения, и для их описания нужны вспомогательные средства. В идеале, было бы сконструировать некоторый математический объект, который можно было бы сопоставить мыслеобразам и использоваться в качестве представления последних.

Одно из таких средств давно известно и широко используется – это модели, или идеализированные представления реальности. В качестве примеров можно привести понятия физического тела или материальной точки из физики. Наиболее строго этот подход развит в теории моделей. Теория моделей [Chang, Keisler 1973] – раздел математической логики, в котором вводится понятие модели как объекта, на котором можно определить истинность логического высказывания (предложения, формулы).

Как известно, формулами в логике высказываний являются сконструированные по определенным правилам последовательности из высказывательных символов $A, B, C, \ldots$, логических связок $\wedge, \vee, \neg$ и скобок $(,)$. Моделями в этом случае являются любые подмножества высказывательных символов. Если мощность множества высказывательных символов равна *n*, то число моделей составит 2^n. При этом количество классов эквивалентных, то есть не различимых на этих моделях, высказываний равно 2^{2^n}.

Логика высказываний является наиболее простой иллюстрацией применения теории моделей. В логике предикатов построение моделей уже существенно сложнее.

В данной работе нас интересует возможность использования подобного подхода при работе с естественным языком. При этом не ставится целью достичь такого же уровня формализации.

Выбор множества моделей определяется предметной областью. Например, если речь идет о зрительных образах и сценах, то моделями естественно выбрать комбинации геометрических фигур. Открытой является проблема построения универсального набора моделей.

Вполне естественно смыслом высказывания назвать множество моделей, на которых оно истинно. Однако для естественного языка истинность не всегда определяется однозначно.

Информация

В логике высказывание либо истинно на модели, либо ложно. Для естественного языка строгая истинность редко имеет место.

Для формализации неопределенности, присущей естественному языку, возможно использовать нечеткую логику или вероятностный формализм. Более адекватным представляется вероятностный подход, поскольку говорящий, как правило, вкладывает в высказывание вполне определенный смысл, который нам неизвестен. Нечеткие множества были бы адекватны, если бы в высказывание вкладывалось несколько смыслов одновременно, с разной степенью выраженности. Аргументом в пользу использования вероятности будет тот факт, что вероятность является фундаментальным понятием, используемым для описания физического мира (в квантовой механике, статистической физике).

Чтобы можно было определить вероятностную меру, на множестве моделей формально должна быть задана σ–алгебра событий. Это требование не создает проблем, поскольку человек различает лишь конечное число градаций чего бы то ни было, поэтому множество моделей всегда можно взять конечным.

Под информацией будем понимать распределение вероятностей на множестве моделей. Смыслом высказывания назовем его функцию правдоподобия на множестве моделей. Используя функцию правдоподобия, можно от априорного распределения перейти к апостериорному (см. раздел о согласовании экспертных высказываний).

В случае, когда высказывание является логической формулой, а множество всех моделей конечно, мера сходства высказываний может быть [Викентьев, 2004] введена как отношение числа моделей, на которых высказывания либо одновременно истинны, либо одновременно ложны, к общему числу моделей.

В общем случае сходство высказываний определяется через сходство их функций правдоподобия. Последнее может вводиться на основе известных способов задания расстояний на распределениях. Пожалуй, наиболее известным из таких расстояний является энтропийная метрика Кульбака. Однако данное расстояние не учитывает, что на самих моделях может быть определено понятие близости, поэтому более подходящей представляется, например, транспортная метрика Монжа-Канторовича.

Интеллект

Одним из признанных видов интеллектуальной деятельности является логический вывод. Поскольку мы распространяем логические термины на всю деятельность сознания, естественно будет определить интеллект путем обобщения понятия логического вывода.

Назовем интеллектом способность интерпретировать высказывания на естественном языке в универсуме человеческого восприятия и выполнять логический вывод. Напомним, что для этого не обязательно обладать человеческим восприятием, достаточно оперировать моделями некоторого универсума, эквивалентного человеческому, поэтому данное определение подходит как для естественного, так и для искусственного интеллекта.

Однако человек оперирует не формальным языком, а образами (которые могут иметь разный уровень абстрагированности), то есть фактически моделями. Известно, что логический вывод можно осуществлять не только путем эквивалентных преобразований высказываний в соответствии с заданными правилами, но и путем установления эквивалентности высказываний на множестве моделей. Именно второй способ, видимо, и присущ естественному интеллекту, и его стоит использовать при моделировании последнего.

Воля и творчество

В известном тесте Тьюринга искусственный интеллект будет признан состоятельным, если в общении он будет неотличим от человеческого.

Но кроме способности поддерживать беседу, человеческий интеллект характеризуют творческие способности.

Очевидно, что творчество – наиболее трудноформализуемая составляющая интеллектуальной деятельности, даже если под формализацией понимать лишь строгое определение. Вполне возможно предположить, что творчество является лишь эффектом проявления определенных свойств используемых человеком моделей. Так, например, интуицию можно объяснить подсознательным манипулированием моделью предметной области, аналогично тому, как компьютер просчитывает динамические модели объекта. Однако возможно существование еще одной составляющей творчества, которую будем называть волей. Заметим, что в данном разделе воля определяется как некоторая логическая возможность, и не обсуждается вопрос доказательства ее реального существования.

Философское исследование понятия воли обычно проводится в отношении ее свободы [Шопенгауэр]. В определении, которое здесь будет предложено, воля и свобода фактически являются синонимами.

Сначала определим волю для элементарных частиц. Современная физическая теория постулирует невозможность точного предсказания в пределах области квантовой неопределенности. Определим волю как нечто, что конкретизирует исход в пределах области определения волновой функции.

Это же определение годится и для воли человека. Для пояснения рассмотрим пример.

Предположим, что мы усилитель теплового шума вмонтировали в робота, который этим шумом и управляется: каждую секунду смотрится значение напряжения и, если оно больше ноля, то робот

поворачивает направо, в противном случае он поворачивает налево. В промежутках между поворотами робот двигается в текущем направлении с некоторой скоростью.

Согласно постулату квантовой механики траектория движения этого робота будет в принципе непредсказуемой, то есть никакие дальнейшие открытия физических законов не дадут ключа к предсказанию его действий, которые абсолютно случайны, то есть не зависят ни от чего из того, что нам известно или дано в ощущение.

Возникает вопрос, не могут ли сходные эффекты иметь отношение к поступкам человека? Современные знания по нейроанатомии и нейрофизиологии не исключают возможность того, что решения субъекта существенно зависят от чисто квантовых эффектов в микромире нервных клеток. Такая возможность кажется вполне правдоподобной, при этом она даёт физическое содержание понятию воля.

В приведенном определении воля является антагонистом мотивов поведения (инстинкта, привычек). В этом смысле воля противоположна желаниям (их рациональной составляющей) и характеру (как совокупности типичных для заданного индивидуума мотивов).

С физической точки зрения мотивам можно сопоставить волновую функцию, определяемую текущей конфигурацией нервных связей и импульсов. Возможно, такая разновидность мотивов, как инстинкт, отражена в самой структуре мозга (в порядке сцепления нейронов и в их составе). Глубоко укоренившиеся привычки так же, наверное, могут закрепляться в структуре. Но мотив — это ещё и образ (представление), который, по-видимому, соответствует текущим электрохимическим импульсам.

Итак, можно резюмировать, что мотивам на физическом уровне соответствует сама структура мозга и вся совокупность электрохимических импульсов, которые в нём протекают. На долю воли при этом остаются квантовые эффекты, присутствие которых мы считаем неизбежным при работе мозга. Основной вопрос этого раздела состоит в том, есть ли принципиальное отличие волевых актов человека от действий "шумящего" робота.

С точки зрения квантовой механики различия нет: как одно, так и другое являются абсолютно случайными процессами. При этом вопрос, как же все-таки реализуется исход при заданной вероятности, просто не рассматривается. Личная позиция автора [Неделько, 1994] заключается в том, что реализация случайного исхода определяется некоторой сущностью, которая может обладать индивидуальностью. И волевые акты человека объединены его личностным единством. Причем эта индивидуальность никак не связана с характером.

В качестве иллюстрации для введенного понятия воли можно привести следующий пример. Рассмотрим программный генератор псевдослучайных чисел в диапазоне $[0, 1]$. Числа, которые он дает, строго говоря, не являются случайными. Однако с точки зрения обычных статистических критериев они ничем не отличаются от случайных. Теперь мы можем рассмотреть два подобных генератора (использующие различные алгоритмы). Анализируя последовательности чисел, ими произведенные, обычными статистическими методами нельзя определить, каким генератором какое число произведено. Тем не менее, от этого они не перестают быть разными генераторами. Точно так же случайность квантовых эффектов вовсе не исключает возможность того, что эти эффекты на самом деле псевдослучайны, и эта псевдослучайность различна в разных ситуациях даже при одинаковых распределениях вероятностей (волновых функциях).

Выясним теперь, какое отношение воля может иметь к интеллекту. Поскольку воля – это «индивидуальный способ реализации случайности», она может, гипотетически, действовать «против энтропии» и, вообще говоря, «не подчиняться закону больших чисел» (смысл этой взятой в кавычки фразы уточняется в следующем разделе). Такое свойство воли может быть (если имеет место) определяющим для творчества, а именно, давать возможность находить решения, которые маловероятно обнаружить случайным поиском. Видимо, в этом аспекте искусственный интеллект будет принципиально уступать естественному.

Содержательная интерпретация вероятности.

Вероятностью называют значение вероятностной меры, которая в свою очередь определяется как некоторая функция, заданная на σ–алгебре событий и удовлетворяющая аксиомам Колмогорова.

При этом теория не дает правил для задания вероятностей в практических ситуациях. Так например, вероятность падения монеты гербом можно выбрать любым числом в интервале $(0, 1)$. Даже если в качестве эксперимента проведено 1000 подбрасываний монеты, из которых 508 реализовались гербом, это не исключает возможности для вероятности быть равной, например, 0,7. Конечно, при такой гипотезе вероятность получить подобный результат эксперимента (такое же или меньшее число гербов) очень низка, но, тем не менее, ненулевая. Хотя даже нулевая вероятность события еще не означает, что событие невозможное.

Поэтому правильное задание вероятностей делается на основе не математических, а эмпирических законов. Основной эмпирический факт в теории вероятностей можно сформулировать следующим образом: в практических задачах возможно задать вероятностную меру так, что более ожидаемыми будет правильно считать более вероятные события.

При этом есть общие правила для адекватного задания вероятностей. Например, если исходы полностью симметричны (или однородны) из физических условий (например, рассматривается симметричная монета), то они должны приниматься равновероятными. Нулевая вероятность события равносильна тому, что это событие гарантированно не произойдет. При этом важно отметить, что даже если событие гарантированно не произойдет, это не значит, что оно является невозможным. Например при случайном выборе числа из равномерного распределения на интервале $[0, 1]$ можно быть уверенным, что результат не будет равен 0,5 (как и вообще любому наперед заданному значению). Однако событие выбора 0,5 не является невозможным.

Теперь укажем, в каком смысле говорилось, что воля может нарушать закон больших чисел. Разумеется, что, являясь теоремой, ЗБЧ не может нарушаться в математическом смысле, а под «нарушением» имелось в виду невыполнение именно эмпирических правил интерпретации вероятностей.

Пусть мы достоверно узнали, что вероятность выигрыша на лотерейный билет А равна 0,8, а на билет В – только 0,2. Очевидно, правильным решением будет сделать ставку на А. Если же реализация исхода определяется человеческой волей, правильным может быть ожидание события с меньшей вероятностью.

Можно было бы считать, что воля изменяет вероятности – но это не оправдано, поскольку это изменение вероятностей не выявляется статистическими средствами. Минимально достаточным будет допущение возможности для воли нарушать вероятностные предпочтения.

Согласование экспертных высказываний.

В качестве примера построения и использования полного пространства моделей для высказываний рассмотрим разработанный ранее подход к построению решающей функции на основе несогласованных вероятностных логических высказываний экспертов [Лбов, Неделько, 1997], [Неделько, 2000].

Будем использовать экспертную информацию, заданную вероятностными логическими высказываниями вида: "**Если** (*температура воздуха в* 13^{00}) ≥ 12°C **и** (*температура воздуха в* 23^{00}) ≥ 7°C **и** (*атмосферное давление*) > 755 мм. р. ст. **или** (*температура воздуха в* 23^{00}) ≤ 4°C, **то** (*заморозок*) с вероятностью 0,4; **степень доверия** высказыванию = 0,8 ".

Смысл такого рода высказываний заключается в оценке зависимости некоторой целевой переменной Y от измеряемых переменных $X_1,\ldots, X_n$. Поэтому моделями будет множество функций в некотором пространстве.

Множество допустимых значений переменной будем обозначать так же, как саму переменную и обозначим:

$$X = \prod_{j=1}^{n} X_j\,,\ Y = \prod_{j=1}^{m} Y_j\,,\ D = X \times Y\,.$$

Пусть в D определена вероятностная мера $P_c[D]$ (квадратные скобки будем использовать, чтобы отличать меру на множестве от $P(\cdot)$ – вероятности события). Для идентификации различных вероятностных мер введем множество C, элементы которого будем называть стратегиями природы и обозначать c.

Определим функцию условной вероятности как $f_c(x) = P_c(y = 1/x)$ – вероятность принадлежности первому классу при условии известного *x*. Формально, $f_c(x) = \frac{d\,P_c^1[X]}{d\,P_c[X]}$, где мера $P_c^{\omega}[X]$ определяется как $\forall E \subseteq X$, $P_c^{\omega}(E) = P_c(E \times \{\omega\})$, а $P_c[X] = P_c^1[X] + P_c^2[X]$ – маргинальная мера, соответствующая $P_c[D]$.

Оценить $f_c(x)$ на основе эмпирической информации *v*, зная ее функцию правдоподобия, можно, определив апостериорное распределение на стратегиях C, используя формулу Байеса:

$$P[C/v] = \frac{P(v/c)P[C]}{\int\limits_C P(v/c)\,d\,P[C]},$$

где $P(v/c)$ – функция правдоподобия для набора высказываний $v = \left\{B_i \,\middle|\, i = \overline{1,N}\right\}$.

Основная идея, которая используется при восстановлении $f_c(x)$ по экспертной информации, заключается в интерпретации появлений экспертных высказываний, как случайных событий, вероятность которых зависит от того, какая *c* имеет место в действительности. Если высказывания сделаны независимо, то $P(v/c) = \prod_{i=1}^{N} P(B_i/c)$.

Для применения статистического подхода необходимо знать, как вероятность появления заданного высказывания зависит от *c*. При этом, если пользоваться формулой Байеса, то искомую зависимость $P(B_i/c)$ достаточно знать с точностью до множителя, не зависящего от *c*. Такую зависимость естественно называть функцией правдоподобия.

Философские вопросы.

Хотя основное содержание данной работы заключается в формулировании некоторого терминологического аппарата, необходимо хотя бы кратко коснуться его философской интерпретации. В качестве эталонного изложения философских концепций будем опираться на классический учебник [Russell, 1946].

Основной вопрос данного раздела: каким из рассмотренных понятий соответствуют сущности в реальности, и какова их подчиненность (какие первичны, какие вторичны).

Человек идентифицирует свое «Я» как сознание, которое выражается в восприятии окружающего мира. Поскольку независимого определения окружающего мира мы не давали, будем пока отождествлять его с совокупностью ощущений в восприятии. Часть этого мира является зависимым от сознания, а именно, человек может управлять своим телом. Подавляющая часть мира не зависит от сознания. Более того, даже зависимая часть имеет ограничения на управление: так человек может переместить свое тело, например, посредством ходьбы, но не телепортации. То, что (хотя бы частично) не зависит ни от чьего сознания, и назовем материей.

Из того, что материя по определению в своих проявлениях не зависит от сознания, еще не следует, что существование материи независимо от существования сознания. И одна из проблем в том, что определить само понятие существования безотносительно сознания достаточно сложно. Действительно, каждому понятно словосочетание «я существую», так же понятно существование того, что находится в ощущении. Но весьма нетривиально наделить смыслом утверждение «материя существует сама по себе», не проводя аналогий с собственным существованием (которые некорректны, если считать материю неодушевленной). Итак, можно выделить по крайней мере три различных понятия, идентифицируемых словом «существование»: существование «Я», существование ощущений и существование модели, то есть абстракции (например, числа), но ни одно из них не подходит для материи.

Для интерпретации изложенного в данной работе подхода, наиболее подходящей философской концепцией будет считать материю и волю первичными сущностями мироздания, которые не имеют

смысла друг без друга (или являются разными «сторонами» одной сущности, как квантово-волновой дуализм в физике). Тогда сознание будет эффектом взаимодействия воли и материи. А в общем случае взаимодействие воли и материи можно считать определением понятия существования в самом базовом смысле. Такая концепция может считаться материалистической, поскольку сознание здесь вторично. Правда, существование также вторично, при этом сознание является не формой существования, а частным случаем, иначе говоря, существование – это нечто, аналогичное сознанию, но более универсальное.

Введенные понятия имеют практическую значимость и в том, что позволяют проинтерпретировать ряд феноменов. В частности, ситуацию, когда индивид совершает так называемое «волевое усилие», можно объяснить тем, что принимается решение, имеющее низкую вероятность при заданной картине мозговых импульсов. Можно выдвигать и более спорные гипотезы, например возможность для чужой воли непосредственно вмешивается в управление мозговой активностью, что например могло бы быть альтернативным объяснением феномена гипноза. Подобная гипотеза, однако, не имеет научных обоснований и отмечена лишь как логическая возможность. Также понятие воли позволяет провести принципиальное различие между понятиями «я хочу» и «мне хочется», то есть между волей и характером, что имеет практическое социальное значение (хотя не дается ответа, как отличить собственную волю от приобретённых или врожденных мотивов, в частности, стереотипов). Свобода для воли в этом контексте означает не отсутствие необходимости, а наличие индивидуальности.

Кроме того, согласно введенному определению воля бессмертна (точнее говоря, понятие смерти к ней неприменимо). Естественно, что это не имеет ничего общего с теорией «переселения душ», поскольку допускается лишь то, что два последовательно живущих индивида «управляются одним и тем же генератором псевдослучайных чисел», но не передача какой-либо информации между ними.

Заключение

Основная идея данной работы заключается в формализации понятия смысла высказывания посредством введения пространства моделей. В частности это позволяет для анализа текстов на естественном языке выбрать математический аппарат, который был бы достаточно строгим, но более гибким и наглядным по сравнению с теориями формальных языков.

Поскольку использование языковых средств является важнейшим атрибутом интеллектуальной деятельности, естественным образом в рассмотрение оказался вовлечен широкий круг сопутствующих понятий.

Литература

[Chang, Keisler 1973] C.C. Chang, H.J. Keisler. Model Theory. / Studies in Logic and Foundations of Mathematics. Vol. 73. London. 1973. – Г. Кейслер, Ч.Ч. Чен. Теория моделей. М.: Мир. 1977. 612 с.

[Викентьев, 2004] А.А. Викентьев. Метрика и информативность на знаниях экспертов в различных моделях теорий // Искусственный интеллект т.2, НАН Украины, 2004, с.37-42.

[Лбов, Неделько, 1997] Г.С. Лбов, В.М. Неделько. Байесовский подход к решению задачи прогнозирования на основе информации экспертов и таблицы данных. // Доклады РАН. Том 357. № 1. 1997. С 29–32.

[Неделько, 2000] В.М. Неделько. Байесовская стратегия прогнозирования разнотипного временного ряда на основе выборки и экспертных высказываний. // III Международная конференции по мягким вычислениям и измерениям (SCM-2000). Сборник докладов. Июнь 2000, С. Петербург. С. 123–126.

[Шопенгауэр] А. Шопенгауэр. Избранные произведения. М. Просвещение, 1992.

[Неделько, 1994]. Проблема свободы воли в философии А. Шопенгауэра. 1994. 12 с.

[Russell, 1946] B. Russell. History of Western Philosophy. London. 1946. – Б. Рассел. История западной философии. Новосибирск. 2003. 991 с.

Информация об авторе

Виктор Михайлович Неделько *– с.н.с. лаборатории Анализа данных Института математики СО РАН, пр-т Коптюга, 4, Новосибирск, 630090, Россия, e-mail: nedelko@math.nsc.ru*

РАСШИРЕННАЯ МОДЕЛЬ „СУЩНОСТЬ-СВЯЗЬ”: ТИПЫ СУЩНОСТЕЙ СУПЕРКЛАСС И ПОДКЛАСС, ТИП СВЯЗИ СУПЕРКЛАСС/ПОДКЛАСС

Дмитрий Буй, Людмила Сильвейструк

***Аннотация**: Рассматриваются и формализуются такие основные понятия расширенной модели „сущность-связь”: тип сущности суперкласс, тип сущности подкласс, тип связи суперкласс/подкласс, тип связи isa. В терминах теории отношений уточняются ограничения, которые накладываются на тип сущности суперкласс и его типы сущностей подкласс.*

***Ключевые слова**: тип сущности суперкласс, тип сущности подкласс, тип связи суперкласс/подкласс, тип связи isa.*

***ACM Classification Keywords**: E.4 Coding and information theory – Formal models of communication.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

В работах [Буй, Сильвейструк, 2006; Буй, Сильвейструк, 2007] рассматривались и формализовались основные понятия модели „сущность-связь” (entity-relationship model): типы сущностей, типы связей, ограничения типов связей, роли, сильные и слабые типы сущностей, сильные и слабые типы связей.

В 80-х годах понятий модели „сущность-связь” стало недостаточно для того, чтобы создавать модели данных. Такая ситуация послужила стимулом к разработке дополнительных концепций семантического моделирования. Таким образом, модель „сущность-связь” была пополнена дополнительными концепциями и получила название *расширенной модели „сущность-связь”* (*Enhanced Entity-Relationship model*) или *EER-модели*.

Расширенная модель „сущность-связь включает все концепции модели „сущность-связь”, а также собственные концепции. Для расширенной модели „сущность-связь”, как и для модели „сущность-связь”, не существует единого общепринятого стандарта, но имеется набор общих конструкций, которые лежат в основе большинства вариантов (интерпретаций) модели.

В данной работе рассматриваются и уточняются (на основе теории отношений) такие понятия этой модели: тип сущности суперкласс, тип сущности подкласс, тип связи суперкласс/подкласс, тип связи isa, согласно [Гарсиа-Молина, Ульман, Уидом, 2004, гл. 2; Дейт, 1998, часть III, гл. 14; Коннолли, Бегг, Страчан, 2003, часть III, гл. 12; Крёнке, 2003, часть II, гл. 3; Elmasri, Navathe, 2004, часть IV].

Следует отметить, что понятия расширенной модели „сущность-связь” не имеют общепринятой и четкой интерпретации; более того, существует существенный терминологический разнобой; поэтому в табл. 1 приведены варианты, встречающиеся в русскоязычной литературе [Гарсиа-Молина, Ульман, Уидом, 2004; Дейт, 1998; Коннолли, Бегг, Страчан, 2003; Крёнке, 2003].

Такой терминологический разнобой может быть обусловлен неточностями перевода соответствующих источников. Если придерживаться терминологии, которая была принята в работах [Буй, Сильвейструк, 2006; Буй, Сильвейструк, 2007], целесообразно суперкласс называть типом сущности суперкласс, подкласс – типом сущности подкласс, связь суперкласс/подкласс – типом связи суперкласс/подкласс, связь isa – типом связи isa.

Табл. 1 – Ключевые понятия расширенной модели „сущность-связь” и их названия

Понятие	1 вариант	2 вариант	3 вариант	4 вариант
Суперкласс	Супертип	Базовый класс	Суперкласс	Надтип
Подкласс	Подтип	Подкласс	Подкласс	Подтип
Связь суперкласс/подкласс	–	–	Связь суперкласс/подкласс	–
Связь isa	Связь isa	Связь isa	–	Связь типа „Есть”
Источник	[Дейт, 1998, часть III, гл. 14]	[Гарсиа-Молина, Ульман, Уидом, 2004, гл. 2]	[Коннолли, Бегг, Страчан, 2003, часть III, гл. 12]	[Крёнке, 2003, часть II, гл. 3]

Типы сущностей суперкласс и подкласс

Понятия типов сущностей суперкласс и подкласс были добавлены в расширенную модель „сущность-связь ” после публикации работы Дж. Смита и Д. Смита [Smith, Smith, 1977]. Причиной стали случаи, когда тип сущности содержит определенные сущности, имеющие специальные свойства, которые не имеют другие сущности данного типа. Поэтому полезно разделять такой тип сущности на несколько специальных типов сущностей, каждый из которых называется *типом сущностей подкласс* (*subclass entity type*), а исходный тип сущности называется *типом сущности суперкласс* (*superclass entity type*).

Тип сущности суперкласс – тип сущности, который включает одну или несколько разных вспомогательных совокупностей его сущностей, которые должны быть представленный в модели данных.

Тип сущности подкласс – вспомогательная совокупность сущностей некоторого типа сущности (типа сущности суперкласс), которая должна быть представлена в модели данных.

Уточнение одного и того же типа сущности может проводиться на основе разных отличительных особенностей, то есть один тип сущности суперкласс может иметь несколько совокупностей типов сущностей подкласс, которые (совокупности) отображают разные способы группировки сущностей исходного типа сущности суперкласс.

Каждая сущность типа сущности подкласс является сущностью соответствующего типа сущности суперкласс, но сущность типа сущности суперкласс не обязательно является сущностью некоторого типа сущности подкласс. Заметим, что тип сущности подкласс должен содержать, как минимум, одну сущность, иначе нет смысла создавать данный тип сущности подкласс. Один и тот же тип сущности не может быть одновременно типом сущности суперкласс и типом сущности подкласс данного типа сущности суперкласс.

Типы сущностей суперклассы и подклассы будем интерпретировать как множества, а сущности указанных типов – как элементы данных множеств. Пусть множество $\mathbf{E}$ соответствует типу сущности суперкласс, а непустые множества $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_n$ – соответствующим типам сущностей подкласс данного типа сущности суперкласс, тогда $\bigcup_{i=1}^{n} \mathbf{E}_i \subseteq \mathbf{E}$.

Далее рассматривается тип сущности суперкласс и соответствующие типы сущностей подкласс, отвечающие некоторому фиксированному способу группировки сущностей суперкласса. При создании типа сущности суперкласс и соответствующих ему типов сущностей подкласс на эти типы сущностей накладываются ограничения. Рассмотрим два основных вида ограничений:

- *ограничение участия* (*participation constraint*);
- *ограничение непересечения* (*disjoint constraint*).

Ограничение участия определяет, каждая ли сущность типа сущности суперкласс относится к некоторому типу сущности подкласс.

Ограничение участия может быть *обязательным* (*mandatory*) или *необязательным* (*optional*). При обязательном участии каждая сущность типа сущности суперкласс должна быть сущностью некоторого типа сущности подкласс. При необязательном участии некоторая сущность типа сущности суперкласс может не быть сущностью ни одного типа сущности подкласс.

Ограничение непересечения описывает связь между сущностями типов сущностей подкласс (которые связаны с одним типом сущностей суперкласс) и указывает, может ли сущность типа сущности суперкласс принадлежать только одному или нескольким типам сущностей подкласс.

Ограничение непересечения может быть *непересекающим* (*disjoint*) или *пересекающим* (*nondisjoint*). В первом случае каждая сущность типа сущности суперкласс может быть сущностью не более одного типа сущности подкласс. Во втором – сущность типа сущности суперкласс может быть сущностью нескольких типов сущностей подкласс.

Понятно, что данное ограничение можно применять, если тип сущности суперкласс имеет более одного типа сущности подкласс.

Так как указанные два ограничения является логически независимыми характеристиками образования типов сущностей суперкласс и подкласс, то при их совместимом использовании выделяют четыре следующих разных вида ограничений:

- обязательное (ограничение участия) и непересекающее (ограничение непересечения);
- необязательное и непересекающее;
- обязательное и пересекающее;
- необязательное и пересекающее.

При обязательном и непересекающем ограничении любая сущность типа сущности суперкласс должна принадлежать одному и только одному типу сущности подкласс данного типа сущности суперкласс.

При необязательном и непересекающем ограничении каждая сущность типа сущности суперкласс либо принадлежит единственному типу сущности подкласс данного типа сущности суперкласс, либо не принадлежит ни одному типу сущности подкласс данного типа сущности суперкласс.

При обязательном и пересекающем ограничении любая сущность типа сущности суперкласс должна принадлежать некоторому типу сущности подкласс (необязательно одному) данного типа сущности суперкласс.

При необязательном и пересекающем ограничении каждая сущность типа сущности суперкласс либо принадлежит некоторому типу сущности подкласс (необязательно одному) данного типа сущности суперкласс, либо не принадлежит ни одному типу сущности подкласс данного типа сущности суперкласс.

Рассматривая обязательное ограничение участия, приходим к теоретико-множественным понятиям покрытия и разбиения множества. Очевидно, что обязательное и пересекающее ограничение можно уточнить с помощью покрытия подмножествами $\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_n$ множества $\mathbf{E}$, то есть $\mathbf{E} = \bigcup_{i=1}^{n} \mathbf{E}_i$ и $\mathbf{E}_i \neq \varnothing$, где $i = 1, 2, \ldots, n$.

Подмножества $\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_n$ образуют разбиение множества $\mathbf{E}$, если рассматривать обязательное и непересекающее ограничение, то есть $\mathbf{E} = \bigcup_{i=1}^{n} \mathbf{E}_i$, $\mathbf{E}_i \neq \varnothing$ и $\mathbf{E}_i \cap \mathbf{E}_j = \varnothing$, где $i \neq j$ и $i, j = 1, 2, \ldots, n$.

Если рассматривать необязательное ограничение участия, то необязательное и пересекающее ограничение можно уточнить посредством покрытия подмножествами $\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_n$ множества $\mathbf{E}$ или его собственного подмножества, то есть $\bigcup_{i=1}^{n} \mathbf{E}_i \subseteq \mathbf{E}$ и $\mathbf{E}_i \neq \varnothing$, где $i = 1, 2, \ldots, n$.

Наконец, необязательное и непересекающее ограничение можно уточнить посредством разбиения подмножествами $\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_n$ множества $\mathbf{E}$ или его собственного подмножества, то есть $\bigcup_{i=1}^{n} \mathbf{E}_i \subseteq \mathbf{E}$, $\mathbf{E}_i \neq \varnothing$ и $\mathbf{E}_i \cap \mathbf{E}_j = \varnothing$, где $i \neq j$ и $i, j = 1, 2, \ldots, n$.

В работе [Буй, Сильвейструк, 2006] ограничения кардинальности, которые накладывались на бинарные типы связей, уточнялись посредством свойств соответствующих отношений (функциональность отношения и включение проекции по первой компоненте отношения в множество, на котором отношение задавалось). Поэтому по аналогии уточняются ограничения непересечения и участия.

Рассмотрим множество $\mathbf{E}' = \mathbf{E} \cup \{\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_n\}$ и зададим на нем бинарное отношение $\prec$: $e \prec \mathbf{E}_i \overset{def}{\Leftrightarrow} e \in \mathbf{E}_i$, то есть множество $\mathbf{E}_i$, $i = 1, 2, \ldots, n$ (тип сущности подкласс) содержит элемент e множества $\mathbf{E}$ (тип сущности суперкласс).

Ограничение непересечения можно уточнить посредством функциональности отношения $\prec$: если отношение $\prec$ функционально, то ограничение непересечения является непересекающим. Если же отношение $\prec$ не функционально, то это ограничение является пересекающим.

Аналогично для ограничения участия: если $\pi_1^2(\prec) = \mathbf{E}$, то ограничение участия обязательное, в случае $\pi_1^2(\prec) \subseteq \mathbf{E}$ ограничения участия необязательное.

В табл. 2 показаны все виды ограничений, которые накладываются на тип сущности суперкласс и его типы сущностей подкласс, а также дано уточнения этих ограничений.

Табл. 2 – Уточнения ограничений, которые накладываются на тип сущности суперкласс и его типы сущностей подкласс

	$\prec$ функциональное	$\prec$ не функциональное
$\pi_1^2(\prec) = \mathbf{E}$	Обязательное и непересекающее ограничение (типы сущностей подкласс образуют разбиение типа сущности суперкласс)	Обязательное и пересекающее ограничение (типы сущностей подкласс образуют покрытие типа сущности суперкласс)
$\pi_1^2(\prec) \subseteq \mathbf{E}$	Необязательное и непересекающее ограничение (типы сущностей подкласс образуют разбиение типа сущности суперкласс или его собственного подмножества сущностей)	Необязательное и пересекающее ограничение (типы сущностей подкласс образуют покрытие типа сущности суперкласс или его собственного подмножества сущностей)

Ограничения, которые накладываются на тип сущности суперкласс и его типы сущностей подкласс, применяются для каждого способа группировки сущностей типа сущности суперкласс. Очевидно, что нет смысла применять данные ограничения к типам сущностей подкласс разных логически независимых способов группировки сущностей типа сущности суперкласс, так как в данном случае ограничения не несут полезную информацию.

Рассматривая тип сущности суперкласс и его типы сущностей подкласс, интересно рассмотреть способы, которыми они могут задаваться. В литературе (смотри, например, [Elmasri, Navathe, 2004, часть IV, с. 103]) предлагается способ *определяющего предиката* (*defining predicate*) и *определяющего атрибута* (*defining*

attribute). То есть для каждого фиксированного способа группировки сущностей типа сущности суперкласс выделяют некоторые определяющие атрибуты, а для каждого типа сущности подкласс некоторый определяющий предикат, заданный на этих определяющих атрибутах (точнее на значениях атрибутов). Для того, чтобы определить принадлежность сущности типа сущности суперкласс соответствующему типу сущности подкласс необходимо проанализировать значение предиката данного типа сущности подкласс: если значение истинно, то принадлежность имеет место, если ложно, то не имеет.

Возникает задача: можно ли задать некоторое отношение на множестве, интерпретирующем тип сущности суперкласс, и по нему получить множества, интерпретирующие типы сущностей подкласс.

Данная задача имеет решение для частного случая – для типа сущности суперкласс и его типов сущностей подкласс с обязательным и непересекающим видом ограничений, которое на них накладывается. Как отмечалось ранее, данный вид ограничений можно уточнить с помощью разбиения. Основываясь на классических теоретико-множественных результатах, отношение эквивалентности позволяет однозначно разбивать множество, на котором задано отношение, на непересекающие классы, содержащие эквивалентные между собой элементы (смотри, например, [Шрейдер, 1971]). Эти классы эквивалентности и интерпретируют типы сущностей подкласс.

При этом следует отметить, что при построении типов сущностей подкласс некоторого типа сущности суперкласс проектант в модели явно указывает количество соответствующих типов сущностей подкласс. В то время как отношение эквивалентности, в общем случае, не позволяет задавать количество необходимых классов эквивалентности (точнее говоря, оценить мощность фактор-множества).

Перейдем от частного случая к более общему, то есть снимем условие транзитивности и обобщим понятие эквивалентности до понятия толерантности [Шрейдер, 1971]. Хорошо известно, что между отношениями толерантности и покрытиями множества существует тесная связь (напомним, что с помощью покрытия выше уточнялся тип сущности суперкласс и его типы сущностей подкласс с обязательным и пересекающим видами ограничений). Однако на этом пути возникает проблема при построении классов толерантности (элементов покрытия): если по покрытию отношение толерантности строится однозначно, то обратная задача построения покрытия по отношению толерантности имеет, вообще говоря, несколько решений [Шрейдер, 1971]. Таким образом, при задании покрытий отношениями толерантности нужны дополнительные средства.

Типы связи суперкласс/подкласс и isa

Тип связи суперкласс/подкласс (*superclass/subclass relationship type*) – это тип связи между типом сущности суперкласс и его типами сущностей подкласс.

Некоторые авторы данный тип связи называют типом связи класс/подкласс (*class/subclass relationship type*) [Elmasri, Navathe, 2004, часть IV, с. 87].

Данный тип связи имеет такие особенности:

- он является $(n+1)$-арним типом связи (то есть это тип связи между одним типом сущности суперкласс и его n типами сущностей подкласс), где $n \geq 1$;
- мор простой кардинальности всех типов сущностей подкласс в этом типе связи суперкласс/подкласс обязательное (по терминологии работы [Буй, Сильвейструк, 2006] степень участия каждого типа сущности подкласс в типе связи суперкласс/подкласс полная (mantandory)).

Тип связи isa (isa relationship type) – это тип связи суперкласс/подкласс между типом сущности суперкласс и его типом сущности подкласс.

Данный тип связи имеет такие особенности:

- он является бинарным типом связи „один к одному” (1:1);
- мор простой кардинальности типа сущности подкласс в этом типе связи обязательное.

Следуя работе [Буй, Сильвейструк, 2006], бинарный тип связи уточняется посредством бинарного отношения, причем бинарный тип связи будет вида „один к одному", когда соответствующее отношение и обратное к нему отношение функциональное. Тип связи isa можно уточнить посредством тождественного отношения, в данном случае вида $\Delta_{\mathbf{E}_1} = \{< e, e > | \, e \in \mathbf{E}_1\}$ (диагональ на множестве $\mathbf{E}_1$). Тривиально, что тождественное отношение и обратное к нему (которое совпадает с исходным отношением) являются функциональными, то есть тип связи isa – это действительно бинарный тип связи „один к одному".

Отметим, что в доступной русскоязычной литературе нет четко очерченных данных типов связей [Гарсиа-Молина, Ульман, Уидом, 2004; Дейт, 1998; Коннолли, Бегг, Страчан, 2003; Крёнке, 2003], причем в [Коннолли, Бегг, Страчан, 2003] употребляется термин „тип связи суперкласс/подкласс" для типа связи isa.

Выводы

Типы сущностей суперкласс и подкласс целесообразно вводить в модель „сущность-связь:

- во-первых, они позволяют не описывать несколько раз аналогичные сущности, благодаря чему экономится время проектировщика, а диаграммы сущностей и связей становятся более удобными для восприятия;
- во-вторых, они позволяют вводить в модель значительный объем семантической информации в форме, приемлемой для пользователей модели.

Литература

[Elmasri, Navathe, 2004] Elmasri R., Navathe S. Fundamentals database systems, 4-th edition. – Pearson: Addison-Wesley, 2004. – 1030 p.

[Smith, Smith, 1977] Smith, J., and Smith, D. Database abstractions: Aggregation and generalization // ACM Transactions on Database Systems. – 1977. – P. 105-133.

[Буй, Сильвейструк, 2006] Буй Д.Б., Сильвейструк Л.Н. Формализация структурных ограничений в модели „сущность-связь" // Proceedings of the XII-th International Conference „Knowledge-Dialogue-Solution" (June 20-25, 2006, Varna, Bulgaria). – Sofia. – 2006.– P. 223-229.

[Буй, Сильвейструк, 2007] Буй Д., Сильвейструк Л. Модель „сущность-связь": роли, сильные и слабые типы сущностей и типы связей // Proceedings of the XIII-th International Conference „Knowledge-Dialogue-Solution" (June 18-24, 2007, Varna, Bulgaria). – Sofia. – 2007. – Vol. 1. – P. 316-322

[Гарсиа-Молина, Ульман, Уидом, 2004] Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс.: пер. с англ. – Москва: Издательский дом „Вильямс", 2004. – 1088 с.

[Дейт, 1998]. Дейт Дж. Введение в системы баз данных.: пер. с англ. – Киев: „Диалектика", 1998.– 784 с.

[Коннолли, Бегг, Страчан, 2003] Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 3-е изд.: пер. с англ. – Москва: Издательский дом „Вильямс", 2003.– 1440 с.

[Крёнке, 2003] Крёнке Д.Теория и практика построения баз данных. 8-е изд. Санкт-Петербург: „Питер", 2003. – 800 с.

[Шрейдер, 1971] Шрейдер Ю.А. Равенство, сходство, порядок – Москва: Наука, 1971. – 255 с.

Информация об авторах

Буй Дмитрий *– заведующий лабораторией проблем программирования, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики: Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: buy@unicyb.kiev.ua*

Сильвейструк Людмила *– аспирантка, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики: Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: slm-klm@rambler.ru*

«МНОЖЕСТВА И РАССТОЯНИЯ СООТВЕТСТВИЯ» В ЗАДАЧАХ КЛАСТЕРИЗАЦИИ: ГИПЕРПЛОСКОСТИ

Николай Кириченко, Владимир Донченко

***Аннотация**: Рассматриваются общие проблемы кластеризации. Предложена концепция «множеств» и «расстояний соответствия» в построении кластеров, рассмотрены модели кластеризации, в которых «множествами соответствия» являются гиперплоскости, а «расстояниями соответствия» – различные варианты расстояний в связи с соответствующими гиперплоскостями. Развит аппарат псевдообращения по Муру – Пенроузу: приведены рекуррентные формулы возмущения для ортогональных проекторов и R-операторов, связанных с псевдообращением. Рекуррентные формулы возмущения использованы для построения алгебраического варианта Jack Knife'a. Приведена сборка важных для приложений результатов, касающихся псевдообращения.*

***Ключевые слова:** кластеризация, кластеризация по гиперплоскостям, псевдообращение по Муру - Пенроузу, сингулярное представление (SVD), ортогональные проекторы, псевдообращение для возмущённых матриц, преобразование Хока.*

***ACM Classification Keywords**: G.3 Probability and statistics, G.1.6. Numerical analysis: Optimization; G.2.m. Discrete mathematics: miscellaneous.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Вступление

Статья посвящена алгебраическим аспектам задачи кластеризации (см., например, [Kohonen, 2001]) как задачи группирования информации. В дальнейшем будет обсуждаться вопрос о разбиении имеющихся элементов на два класса с тем, что процедуру такого разбиения можно запускать рекурентно.

Важным, как представляется авторам, во всех методах кластеризации является представление о «множествах соответствия» и «расстояниях соответствия». Типичным представителем первых являются прототипы-представители (prototypes) классов в методе к-средних. Что касается «расстояний соответствия», то, это меры соответствия «множествам соответствия», в соответствии с которыми элемент относят к тому или иному классу: как правило, – по минимальному значению «расстояния». Как правило, такими расстояниями соответствия являются, евклидовы расстояния в соответствующих пространствах признаков.

Заметим также, что процедуры кластеризации построены на применении стандартной рекуррентной процедуры: последовательного объединения (merging), разбиения (splitting) или уточняющих друг друга разбиений.

Разделяют также процедуры кластеризации с учителем (обучение с учителем – supervised learning) и – без учителя (unsupervised learning). В первом случае имеющиеся элементы уже разделены на классы, во втором – следует выделить классы на основе анализа внутренней структуры совокупности $x(1),\ldots,x(n)$ векторов из пространства признаков R^m

В задачах кластеризации следует также выделять этап обучения: построения соответствующих классов-кластеров (этап обучения), и этап использования построенного разбиения: отнесения каждого нового вектора признаков к одному из построенных классов.

В предлагаемой вниманию читателю работе речь идёт об использовании гиперплоскостей в качестве «множеств соответствия», о «расстояниях соответствия», построенным в связи с гиперплоскостями, а

также об обеспечении рекуррентности применения процедуры обучения без учителя; о согласованности обучения с учителем и без него; об аппарате псевдообращения по Муру – Пенроузу ([Moore,1920], [Penrose,1955]); о важном продвижении и расширении возможностей аппарата псевдообращения: о теории возмущения псевдообратных матриц ([Кириченко, 1997]), а также – о её совершенствовании и применении в задачах кластеризации (см. также [Кириченко, Донченко, 2007 a),b)]). Заметим, что важные примеры применения теории псевдообращения к исследованию классических прикладных задач, отличных от задач кластеризации, можно найти в работах [Кириченко, Лепеха, 2002], [Кириченко, Донченко, 2005] .

Заметим также, что важными вехами в развитии аппарата псевдообращения, в частности, в обеспечении эффективности построения соответствующих рекуррентных процедур и вычисления расстояний соответствия в них, – являются: прямые [Албert, 1977] и обратные [Кириченко, 1997] формулы Гревиля; формулы псевдообращения для замены строки или столбца матрицы [Кириченко, Лепеха, 2002,], [Кириченко, Донченко, 2005]; также формулы возмущения для Z - и R-операторов [Кириченко, Донченко., 2007 b)]. Отметем также, что задача кластеризации по гиперплоскостям, порождённым пространствами значений подходящих афиннных операторов, как вариант применения преобразования Хока, рассматривалась в работе [Donchenko, 2003].

В первой части предлагаемой работы приводится подборка результатов, важных в технике применения аппарата псевдообращения.

Во второй части рассматривается собственно задача кластеризации: рассматриваются подходящие гиперплоскости в качестве «множеств соответствия», строятся подходящие «расстояния соответствия» в связи с введёнными в рассмотрение гиперплоскостями, рассматриваются проблема обеспечения рекуррентности в вычислении «расстояний соответствия» как внутри рекуррентного шага, так и между разными шагами.

Отметим, что аппарат псевдообращения позволяет выписывать явные формулы, как для «расстояний соответствия», так и явно описывать «множества соответствия» в терминах смещения и явного описания ортогональных проекторов соответствующих линейных подпространств (ср. с вычислительными процедурами [Vapnik,1998] для статистических вариантов кластеризации на основе ковариационных). Вычислительные алгоритмы для расстояния от гиперплоскостей использовались, к примеру, также в работе [Haykin,1999].

Постановка задачи

Собственно, использование гиперплоскостей как аппарата решения задач группирования информации в статистической постановке, восходит к методу главных компонент: [Pearson , 1901] (другие названия метод Хётеллинга (*Hotelling*), метод Карунена-Лоэва (*Karhunen-Loeve*)) и имеет в основе идею такого ортогонального преобразования имеющегося набора случайных величин, которое бы приводило матрицу ковариаций к главным осям. Ещё раз обратим внимание читателя на специфически статистический вариант постановки и применения метода кластеризации в виде метода главных компонент, связанный с анализом естественного матричного объекта, каковым является матрица ковариаций, и применению классического результата Сильвестра [Sylvester, 1889]. Псевдообращение позволяет анализировать матрицы произвольной размерности, а не обязательно квадратные; позволяет эффективно строить ортогональные проекторы, отвечающие «естественным подпространствам» линейного оператора: подпространству значений и ядру оператора; описывать гиперплоскости, отвечающие всем решениям системы линейных алгебраических уравнений (СЛАУ), а также описывать необходимые и достаточные условия существования таких решений: описывать «наилучшие» приближенные решения (псевдорешения) СЛАУ; явно описывать невязку соответствующего приближения.

В последующем будем рассматривать задачу кластеризации в обучении без учителя для дихотомического варианта постановки задачи: для разбиения имеющейся совокупности *x(1),…,x(n)* векторов:из пространства признаков R^m на две части. В качестве множеств соответствия для каждого из классов-кластеров будут рассматриваться две гиперплоскости $\Gamma(k) \subseteq R^m, k = 1,2$: $\Gamma(k) = x_k + L_k \subseteq R^m, k = 1,2$, *x* – будем называть смещение гиперплоскости, . *L* –подпространством гиперплоскости. Таким образом, решение задачи кластеризации в такой постановке включает в себя

- построение «множеств соответствия» в виде гиперплоскостей: описание их смещений и соответствующих подпространств;
- описание «расстояний соответствия»;
- разбиение векторов *x(1),…,x(n)* обучающей выборки на две части в соответствии с минимумом «расстояния соответствия» на две части:

  $x(i_1),\ldots,x(x(i_{n_1})) \in \Gamma(1)$, $x(j_1),\ldots,x(x(j_{n_2})) \in \Gamma(2)$:

  $\{i_1,\ldots,i_{n_1}\} \cup \{j_1,\ldots,j_{n_2}\} = \{1,2,\ldots,n\}$, $n_1 + n_2 = n$;

- построение решающего правила, в соответствии с которым следует относить объект, не представленный в выборке, к одному из двух классов.

Естественным для рекуррентной процедуры построения классов-кластеров является получение и использование результатов, обеспечивающих рекуррентность.

Заметим, что вариантом указанной задачи кластеризации является такой, в котором дополнительно фиксируется общая размерность s: s<m гиперплоскостей $\Gamma(k) = x_k + L_k, k = 1,2$.

Напомним, что «гиперплоскости соответствия» подлежат определению на основе внутренней структуры имеющегося набора векторов *x(1),…,x(n)*.

Вспомогательные определения и утверждения

Псевдообращение и сингулярное (SVD –) представление. Псевдообращение – обозначается A^+ – по Муру - Пенроузу ([[Moore,1920], Penrose,1955], см. также [Алберт, 1977]) для $m \times n$ матрицы *A* может определяться одним из нескольких эквивалентных способов, среди которых отметим определение через сингулярное представление матриц (SVD-разложение), когда псевдообращение определяется соотношением:

$$A^+ = \sum_{i=1}^{r} x_i y_i^T \lambda_i^{-1}, \quad (1)$$

которое определяется элементами SVD-представления исходной матрицы:

$$A = \sum_{i=1}^{r} y_i x_i^T \lambda_i, \quad (2)$$

в котором: $\lambda_1^2 \geq \ldots \lambda_r^2 > 0$ – общий набор ненулевых собственных чисел матриц $AA^T, A^T A$, $y_i, i = \overline{1,r}$ и $x_i, i = \overline{1,r}$, соответственно, – ортонормированные наборы собственных векторов этих матриц, а *r = rank A= rank* A^T.

Чаще всего сингулярное разложение матрицы A представляется в виде, определяемом следующей леммой.

Лемма 1. Для любой $m \times n$ матрицы *A* ранга *r* существуют $Y - m \times r$ и $X - r \times n$ с ортонормированными столбцами и строками соответственно, а также диагональная матрица $\Lambda = diag(\lambda_1,\ldots,\lambda_r), \lambda_1 \geq \ldots \geq \lambda_r > 0$ такие, что

$A = Y\Lambda X$. (3)

Представление (2) является эквивалентным вариантом представления (3), если через $x_i, i = \overline{1,r}$ обозначить столбцы (ортонормированные) матрицы X, а через $y_i^T, i = \overline{1,r}$ – строки (ортонормированные) матрицы Y. В таких обозначениях справедлива следующая лемма.

Лемма 2.Произведение $Y\Lambda X$ матриц в (3) может быть представлено через «столбцовое» для Y и «строчное» для X представление может быть представлено е в виде

$$X\Lambda Y = \sum_{i=1}^{r} y_i x_i^T \lambda_i .$$

Собственно, указанное несложное утверждение вытекает из того, что произведение BC двух матриц со «столбцовым» предоставлением для первой и «строчным» представлением для второй::

$$B = \left(b(1),\ b(2),\ \ldots,\ b(p)\right), \qquad C = \left(c(1),\ c(2),\ \ldots,\ c(p)\right)^T \tag{4}$$

допускает представление в виде

$$BC = \left(b(1),\ b(2),\ \ldots,\ b(p)\right)\left(c(1),\ c(2),\ \ldots,\ c(p)\right)^T = \sum_{i=1}^{p} b(i)c_{(i)}^T \tag{5}$$

В дальнейшем представление (4) будет использоваться и для других матриц. При этом обозначение обозначения $b(i)$, $c_{(i)}^T$, $i = \overline{1,p}$, будут использоваться для обозначения соответственно строк и столбцов необходимых матриц.

Основные ортогональные проекторы: *P*-проекторы, *Z*-проекторы. Псевдообращение позволяет в явном виде выписать пару ортогональных проекторов (ОП), – обозначим их $P(A), P(A^T)$, и будем называть P- проекторами, – на подпространства $L(A^T), L(A)$ значений операторов A^T, A соответственно: $P(A) = A^+A, P(A^T) = A^{T+}A^T = AA^+$. Ортогональные проекторы, которые будем обозначать $Z(A), Z(A^T)$ и называть Z- проекторами – определим соотношениями: $Z(A) = E_n - P(A), Z(A^T) = E_m - P(A^T)$ соответственно. Очевидным образом, Z- проекторы являются ортогональными проекторами на подпространства $L_{A^T}^{\perp}, L_A^{\perp}$, ортогональные к подпространствам $L(A^T), L(A)$ соответственно. Заметим, что $L_{A^T}^{\perp} = KerA$, $L_A^{\perp} = KerA^T$. Соответственно, $Z(A), Z(A^T)$ являются ортогональными проекторами на подпространства нулей $KerA, KerA^T$ операторов A, A^T соответственно.

Замечание 1. Обратим внимание также на то, что каждое из подпространств $L(A), L(A^T)$ является линейной оболочкой соответственно векторов-столбцов и векторов-строк матрицы A.

R-операторы. Важными в связи с определением расстояний соответствия и рекуррентными формулами псевдообращения: формулами позволяющими записывать соответствующий оператор при добавлении или вычёркивании строки или столбца матрицы, – являются также операторы, которые будем называть R - операторы. Их будем определять соотношениями:

$R(A) = A^+A^{T+}, R(A^T) = A^{+T}A^+$.

Важную роль в реализации аппарата псевдообращения в прикладных задачах играют прямые (см., например, [Алберт, 1977]) и обратные [Кириченко 1997] формулы Гревиля(Greville), а также формулы возмущения псевдообращения [Кириченко, 1997]. И в том и в другом случае речь идёт о формулах, связывающих псевдообращение преобразованной матрицы с псевдообращением исходной. В первом случае (прямых или обратных формулах Гревиля) речь идёт о преобразовании матрицы введением или вычёркиванием дополнительного строки или столбца. Во втором – о преобразовании исходной матрицы

аддитивной добавкой ab^T . Таким образом, в формулах псевдообращения для возмущённых матриц речь идёт о выражении псевдообращения возмущенной матрицы $(A+ab^T)^+$ через A, A^+, a, b .

Прямые и обратные формулы Гревиля приведены ниже. Формулы возмущения псевдообращения можно найти в уже цитированной работе [Кириченко 1997]. Ниже приведены полученные на их основе формулы возмущения для *Z*- и *R*-операторов.

Заметим, что комбинация прямых и обратных формул Гревиля позволяет получить формулы псевдообращения при замене строки или столбца исходной матрицы. Соответствующие представления можно найти в работах [Кириченко, Лепеха, 2002], [Кириченко, Донченко, 2005]. Там же приведены формулы, определяющие вид, *Z* - и *R* - операторов при замене строки или столбца матрицы, для которой они рассматриваются.

Прямые формулы Гревиля (Greville).

Напомним, что прямые формулы Гревиля – это формулы, определяющие вид псевдообращении матрицы при её дополнении строкой или столбцом. Они определяются соотношениями, в которых используется блочное представление псевдообращения расширенной матрицы: через $P-m\times n$ - матрицу и, q - $n\times 1$ - вектор для расширения матрицы строкой

$$\begin{pmatrix} A \\ a^T \end{pmatrix}^+ = (P \vdots q) \tag{6}$$

и через $Q-n\times m$ - матрицу и, q - $m\times 1$ - вектор

$$(A \vdots a)^+ = \begin{pmatrix} Q \\ q^T \end{pmatrix} \tag{7}$$

при дополнении матрицы столбцом.

Замечание 2. Обратим внимание читателя, что вектор a в (6) и (7) имеет разные размерности: размерность $n\times 1$ в первом и – $m\times 1$ во втором.

Теорема 1.(прямые формулы Greville– дополнение строкой). В представлении (6)

$$\begin{cases} P=(E-qa^T)A^+ \\ q=\begin{cases} \dfrac{Z(A)a}{a^TZ(A)a}, a^TZ(A)a>0(\text{нез.}) \\ \dfrac{R(A)a}{1+a^TR(A)a}, a^TZ(A)a=0(\text{зав.}) \end{cases} \end{cases}, \tag{8}$$

Теорема 2(прямые формулы Greville– дополнение столбцом) В представлении (7)

$$\begin{cases} Q==A^+(E-aq^T) \\ q=\begin{cases} \dfrac{Z(A^T)a}{a^TZ(A^T)a}, a^TZ(A^T)a>0(\text{нез.}) \\ \dfrac{R(A^T)a}{1+a^TR(A^T)a}, a^TZ(A^T)a=0(\text{зав.}) \end{cases} \end{cases}, \tag{9}$$

Замечание 3. Вид вектора q в прямых формулах Гревиля определяется линейной зависимостью вводимого вектора a^T или a от, соответственно, строк или столбцов матрицы A. Линейная независимость обеспечивается нулевым значением квадратичной формы (с соответствующей матрицей - Z–оператором) на векторе a .

Обратные формулы Гревиля. Как и в прямых формулах Гревиля, вид выражений, связывающих псевдообращения исходной и преобразованной матрицы, выписывается в рамках блочного представлением (6) или (7), и так же – определяется линейной зависимостью или независимостью вычёркиваемой строки или столбца: сохранением или падением ранга преобразованной матрицы.

Изменяется только вид соответствующего условия. Теперь условием независимости является условие $a^T q = 1$.

Теорема *3.* (обратные формулы Гревиля – вычёркивание строки) В обозначениях (6) имеет место соотношение

$$A^+ = \begin{cases} \left(I_n - \dfrac{qq^T}{\| q \|^2} \right) P, a^T q = 1, (\text{нез.}), & \text{ранг падает} \\ \left(I_n - \dfrac{qa^T}{1 - a^T q} \right) P, a^T q < 1 (\text{зав.}), & \text{ранг сохраняется} \end{cases} . \qquad (10)$$

Теорема *4.* (обратные формулы Гревиля – вычёркивание столбца) В обозначениях (7) имеет место соотношение

$$A^+ = \begin{cases} Q\left(I_m - \dfrac{qq^T}{\| q \|^2} \right), a^T q = 1, (\text{нез.}), & \text{ранг падает} \\ Q\left(I_m - \dfrac{aq^T}{1 - a^T q} \right), a^T q < 1 (\text{зав.}), & \text{ранг сохраняется} \end{cases} .$$

Теорема 5 (формулы возмущения для *Z*- и *R*- операторов). При возмущении матрицы A матрицей $a \times b^T$ *Z*- и *R*- операторы для возмущённой матрицы определяется следующими соотношениями, вид которых определяется линейной зависимостью или независимостью векторов - составляющих возмущения от соответствующих составляющих матрицы A, а также от того, сохраняется или падает ранг возмущённой матрицы:

1) Для векторов a и b^T линейно не зависимых от, соответственно, столбцов и строк матрицы A, т.е. при выполнении условий $a^T Z(A^T)a > 0, b^T Z(A)b > 0,$ справедливы следующие соотношения

$$Z(A + ab^T) = Z(A) + \frac{Z(A)bb^T Z(A)}{b^T Z(A)b};$$

$$Z((A + ab^T)^T) = Z(A^T + ba^T) = Z(A^T) + \frac{Z(A^T)aa^T Z(A^T)}{a^T Z(A^T)a};$$

$$R(A + ab^T) = R(A) - R(A)\frac{bb^T Z(A)}{b^T Z(A)b} \quad \frac{Z(A)bb^T}{b^T Z(A)b} R(A) \quad cA^+ ab^T Z(A) \quad cZ(A)ba^T A^{+^T} +$$

$$+ \frac{A^+ aa^T A^{+^T}}{a^T Z(A^T)a} + \frac{b^T R(A)ba^T Z(A^T)a + \left(1 + b^T A^+ a\right)^2}{a^T Z(A^T)a\left[b^T Z(A)b\right]^2} Z(A)bb^T Z(A),$$

где $c = \dfrac{1 + b^T A^+ a}{a^T Z(A^T)ab^T Z(A)b}$.

2) Для вектора a линейно зависимого от столбцов матрицы *A*, а вектора b^T – линейно не зависимого от строк матрицы таким образом, что, – для упрощения представления результата, – $b \perp L_{A^T}$, т.е. при выполнении условий $a^T Z(A^T)a = 0, b^T Z(A)b = \| b \|^2$, справедливы соотношения:

$$Z(A + ab^T) = Z(A) + \frac{k_{A,a,b} k_{A,a,b}^T}{\| k_{A,a,b} \|^2} \quad \frac{bb^T}{\| b \|^2},$$

где: $k_{A,a,b} = A^+ a \quad \frac{b}{||b||^2}$,

$Z((A+ab^T)^T) = Z(A^T + ba^T)$,

$R(A+ab^T) = \left(I_n - \frac{kk^T}{||k||^2}\right) R(A) \left(I_n - \frac{kk^T}{||k||^2}\right)$.

3) Для векторов a и b^T одновременно линейно зависимых от соответственно столбцов и строк матрицы A, при условии падения ранга возмущённой матрицы: $rank(A+ab^T) = rankA - 1$, т.е. при выполнении условий: $a^T Z(A^T)a = 0, b^T Z(A)b = 0, b^T A^+ a = -1$, справедливы следующие соотношения:

$$Z(A+ab^T) = Z(A) + \frac{A^+ aa^T (A^+)^T}{a^T R(A^T)a},$$

$$Z((A+ab^T)^T) = Z(A^T + ba^T) = Z(A) + \frac{(A^+)^T bb^T A^+}{b^T R(A)b},$$

$$R(A+ab^T) = A^+(a,b)A^{+T}(a,b),$$

где: $A^+(a,b) = A^+ \quad \frac{A^+ aa^T R(A^T)}{a^T R(A^T)a} \quad \frac{R(A)bb^T A^+}{b^T R(A)b} + cA^+ ab^T A^+$, $c = \frac{b^T R(A) A^+ a}{a^T R(A^T)a \, b^T R(A)b}$.

4) Для векторов a и b^T одновременно линейно зависимых от, соответственно, столбцов и строк матрицы A, но при условии неизменности ранга возмущённой матрицы по сравнению с рангом A, т.е. при выполнении условий

$a^T Z(A^T)a = 0, b^T Z(A)b = 0, b^T A^+ a \neq -1$,

справедливы следующие соотношения:

$$Z(A+ab^T) = Z(A), \qquad Z((A+ab^T)^T) = Z(A^T + ba^T) = Z(A^T),$$

$$R(A+ab^T) = R(A) \quad \frac{A^+ ab^T R(A)}{1+b^T A^+ a} \quad \frac{R(A)ba^T A^{+T}}{1+b^T A^+ a} + \frac{b^T R(A)b}{1+b^T A^+ a} A^+ aa^T A^{+T}.$$

Основные элементы кластеризации по гиперплоскостям – множества соответствия

Как уже упоминалось, построение «множеств соответствия» в виде гиперплоскостей предполагает конструктивное описание их смещений и соответствующих линейных подпространств.

Смещение гиперплоскостей.

Смещения предлагается определять как средние векторов, принадлежащих к каждой из частей разбиения. Можно также выбрать в качестве смещения один из элементов разбиения.

Подпространства гиперплоскостей. При наличии смещений подпространства гиперплоскостей определяются как подпространства, натянутые на центрированные смещением (преобразованные вычитанием определённого вектора) векторы каждой из частей разбиения. В дальнейшем будет предполагаться, что центрирование каждой части разбиения производится соответствующими средними $\bar{x}_k, k = 1,2$.

Конструктивное описание подпространств, натянутых на каждую из центрированных совокупностей векторов, обеспечивается построением для каждой из гиперплоскости подходящей матрицы $A_k, k = 1,2$ так, чтобы подпространство - множество значений $L_k, k = 1,2$ каждой из них совпадало с подпространством соответствующей гиперплоскости, т.е. с линейной оболочкой каждой из центрированных групп векторов. В соответствии с замечанием 1 в качестве таких матриц можно выбрать матрицы, столбцами которых являются центрированные векторы каждой из частей разбиения

соответственно. В этом случае ортогональными проекторами $P_{L_k}, k = 1,2$ для каждого из подпространств гиперплоскостей будут P - проекторы для транспонированных к соответствующим матрицам:

$P_{L_k} = P(A_k^T), k = 1,2$.

Таким образом, гиперплоскости $\Gamma_k, k = 1,2$ определяются парами $(\bar{x}_k, A_k), k = 1,2$:

$$\Gamma_k = \Gamma(\bar{x}_k, A_k), k = 1,2 \quad (11)$$

Основные элементы кластеризации по гиперплоскостям – расстояния соответствия

В качестве «расстояний соответствия» векторов до каждого «множеств соответствия» $\Gamma_k = \Gamma(\bar{x}_k, A_k), k = 1,2$ предлагается рассматривать евклидово расстояние векторов до гиперплоскостей $\Gamma_k = \Gamma(\bar{x}_k, A_k), k = 1,2$, каковыми эти «множества соответствия» являются. Средства псевдообращения позволяют конструктивно описать соответствующие расстояния. Такое конструктивное описание возможно и в том случае, когда задаётся размерность s : $s \le min(rank A_k, k = 1,2)$ – подпространств гиперплоскостей. Формулы, определяющие соответствующие расстояния, являются предметом следующей леммы.

Лемма 1 . Для $\Gamma_k = \Gamma(\bar{x}_k, A_k), k = 1,2$ расстояния соответствия $\rho(x, \Gamma_k), k = 1,2$ произвольного вектора $x \in R^m$ до каждой из двух гиперплоскостей $\Gamma_k, k = 1,2$ определяются соотношением:

$$\rho(x, \Gamma_k) = (x - \bar{x}_k))^T Z(U_s^T(k))(x - \bar{x}_k), k = 1,2 , \quad (12)$$

Где $U_s(k) = \begin{cases} A_k = \sum_{i=1}^{r} y_i(k) x_i^T(k) \lambda_i(k) & \text{ðàçì .s í å çàäàíà} \\ \sum_{i=1}^{s} y_i(k) x_i^T(k) \lambda_i(k) & \text{ðàçì .s çàäàíà} \end{cases}, r = 1.2 .$

Кластеризации по гиперплоскостям – основные шаги алгоритма

Алгоритм кластеризации по гиперплоскостям состоит в последовательном, рекуррентном уточнении «множеств соответствия», каковыми являются гиперплоскости. На каждом рекуррентном шаге происходит уточнение набора элементов, порождающих «множества соответствия», построение пар $(\bar{x}_k, A_k), k = 1,2$ отвечающих уточнённому разбиению, после чего происходит новое «уточнение разбиения» отбором в каждую часть разбиения векторов исходного набора по минимуму расстояний до вновь построенных гиперплоскостей. В общем, алгоритм состоит в выполнении следующих шагов.

1. На первом шаге производится разбиение на две совокупности произвольным образом.
2. На втором шаге для каждой из частей разбиения вычисляются:
   - смещения $\bar{x}_k, k = 1,2$, как средние по векторам каждой из частей разбиения;
   - матрицы $A_k, k = 1,2$, как матрицы, построенные из центрированных соответствующими средними векторов каждой из групп как из столбцов.

3. На третьем шаге происходит «уточнение» разбиения: вычисляются «расстояния соответствия» каждого из векторов *x(1),…,x(n)* до каждого из двух построенных «множеств соответствия»: до каждой из двух гиперплоскостей, – и происходит отнесение каждого из векторов *x(1),…,x(n)* к той части разбиения, к которой он оказался ближе по «расстоянию соответствия» (12). В результате происходит формирование нового, «уточнённого» разбиения векторов *x(1),…,x(n)* на две части.

4. На четвёртом шагу происходит возвращение ко второму шагу алгоритма.

Кластеризация по гиперплоскостям – модификация расстояний

Расстояния до гиперплоскостей в лемме 1 определяются значениями квадратичных форм с матрицами $\sum_{i=1}^{s} y_i(k)x_i^T(k), k = 1,2$,

Их можно рассматривать как взвешенное среднее матриц $y_i(k)x_i^T(k), i = \overline{1,r}, k = 1,2$ с весами $\omega_i = \begin{cases} 1, i \le s \\ 0, i = \overline{s+1,r} \end{cases}$, соответственно, в нормированном варианте $\omega_i = \begin{cases} 1/s, i \le s \\ 0, i = \overline{s+1,r} \end{cases}$.

Рассмотрение взвешенных варианта сумм из $y_i(k)x_i^T(k), i = \overline{1,r}, k = 1,2$ с нормированными весами $\lambda_i^2(k)$, $i = \overline{1,r}$, $k = 1,2$ даёт следующий вариант расстояний ρ_R до «множеств соответствия». Они для этого случая определяются соотношением:

$$\rho_R(x, \Gamma_k) = \frac{1}{trR(A_k^T(k)A_k)}(x - \bar{x}_k)^T R(A_k^T)(x - \bar{x}_k), k = 1,2 . \quad (13)$$

Использование в качестве «расстояний соответствия» расстояний, определяемых соотношением (13) приводит к очевидному изменению алгоритма кластеризации: в нём «уточнение» разбиения третьего шага происходит на основе «расстояний соответствия», определяемых соотношениями (13) вместо – (12).

Рекуррентные формулы для алгебраического Jack Knife'а

При проверке элементов совокупностей на соответствие вычислением расстояний по формулам (12) или (13) тестируемые элементы принимают участие в формировании гиперплоскостей, представляющих кластеры. Резонной является также построение такой процедура проверки соответствия, при которой тестируемый элемент кластера, исключается из числа объектов, которые его определяют. В статистике такая процедура исключения носит название "Jack Knife"(складной нож) [Эфрон, 1988]. Поэтому процедуру тестирования на принадлежность кластеру с исключением тестируемых элементов из описания кластера будем называть алгебраическим Jack Knife'ом.

Заметим, что естественным является вариант кластеризации, когда исключение элемента приводит к падению ранга матрицы $A(k), k = 1,2$ ((п.3 теоремы 5)). Псевдообращение даёт конструктивную явную формулу проверки соответствующего условия.

Исключение тестируемых элементов из кластера изменяет как сдвиг (центр кластера), так и линейное подпространство кластера. Формулы (12),(13) при таком исключении, очевидным образом, переписываются в виде, для изменённых смещений (будем считать их средними) и изменённых матриц: $x_k^{(0)}, A^{(0)}(k), k = 1,2$ соответственно.

Лемма 1 даёт возможность эффективной организации процедуры «отсеивания», в которой критерий замены строится на основе леммы 1 и имеет вид, определяемый следующей теоремой.

Теорема 6. В условиях падения ранга (п.3 теоремы 5) расстояния до гиперплоскостей, после исключения элемента из числа порождающих элементов, определяется следующим соотношением для одного из значений $k=1$ или $k=2$

$$\rho(x_j(k), \Gamma_j^{(0)}(k)) = \frac{n_k^2}{\left|\left| \left(E_m \quad \frac{q_j(k)q_j^T(k)}{||q_j(k)||^2} \right) \sum_{l \ne j} q_l(k) \right|\right|^2}, j = \overline{1,n_k}, k = 1,2 ,$$

Где $x_j(k)$, $\Gamma_j^{(0)}(k)$ $j=\overline{1,n_k}, k=1,2$ – исключаемые элементы каждой из совокупностей и гиперплоскости, отвечающие «усечённым» совокупностям, а $q_j(k)$, $j=\overline{1,n_k}, k=1,2$ столбцы с номером j, $j=\overline{1,n_k}$ в каждой из матриц $A_k^+, k=1,2$.

Заключение

В работе рассмотрены задачи кластеризации на основе концепции «множеств» и «расстояний соответствия», предложены варианты алгоритмов кластеризации, когда «множествами соответствия» являются гиперплоскости, а «расстояния соответствия» построены на основе вариантов расстояний до них. Применение аппарата псевдообращения позволяет описать все элементы соответствующих построений явными формулами, включая варианты алгебраического Jack Knife'a

Литература

[Алберт, 1977] Алберт А. Регрессия, псевдоинверсия, рекуррентное оценивание. М.: Наука, 1977.–305 с.

[Donchenko, 2003] Donchenko V.S. Hough Transform and Uncertainty//Proceedings International Conference "Knowledge Dialog – Solution". – V. – June 16-23, 2003.–Varna (Bulgaria). – P.391-395.

[Haykin,1999] Neural networks. A comprehensive Foundation. – New Jersey. – 1999.– 842 p.

[Кириченко, 1997] Кириченко Н.Ф. Аналитическое представление псевдообратных матриц//Киб. и СА. - №2, 1997– С.98-122.

[Кириченко, Донченко, 2007, a)] Кириченко Н.Ф., Донченко. В.С. Псевдообращение в задачах кластеризации.// Киб. и СА. - №4, 2007.– С.98-122.

[Кириченко, Донченко, 2007, b)] Кириченко Н., Донченко В. Алгебраический Jack Knife: кластеризация по гиперплоскостям// Proceedings: XIII-th International Conference "Knowledge –Dialog – Solution".–June 18-24, 2007, Varna (Bulgaria). – 2007. – V.1.– P.89-95.

[Кириченко, Лепеха, 2002] Кириченко Н.Ф., Лепеха Н.П. Псевдообратные и проекционные матрицы в применении к исследованию задач управления, наблюдения и идентификации// Киб. и СА.- №4, 2002. – С.107-123.

[Kohonen, 2001] Kohonen T., Self-Organizing Maps.– Third Extended Edition.– New York, 2001.– 501 p.

[Кириченко, Донченко, 2005] Кириченко М.Ф, Донченко В.С. Задача термінального спостереження динамічних системи: множинність розв'язків та оптимізація //Ж. Обч. та пр. мат. – Вип..3 , 2005.– С. 63-78.

[Moore,1920] Moore E.H. On the reciprocal of the general algebraic matrix//Bull. Amer. Math. Soc. – 26, 1920. – P. 394-395.

[Pearson, 1901] Pearson K., On lines and planes of closest fit to systems of points in space//Philosophical Magazine.–1901, N 2.– P. 559—572.

[Penrose,1955] Penrose R. A generalized inverse for matrices// Proc. Cambr. Philosophical Soc.- 51, 1955.– P. 406-413.

[Sylvester, 1889] Sylvester J.J. On the reduction of a bilinear quantic of the nth order to the form of a sum of n products by a double orthogonal substitution, Messenger of Mathematics, – 1889.– N19.– P., 42—46;

[Vapnik, 1998] Vapnik V.N. Statistical Learning Theory.–New York: Wiley. – 1998.

[Эфрон, 1988] Эфрон Б. Нетрадиционные методы многомерного статистического анализа. – М.: Фин. и стат. – 1988.– 263 с.

Информация об авторах

Кириченко Николай Ф.– *Профессор, Институт кибернетики им. В.М.Глушкова НАН Украины, ведущий научный сотрудник.*

Донченко Владимир С. – *Профессор, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, Украина, e-mail: voldon@unicyb.kiev.ua*

ТРЕХЗНАЧНЫЕ ЛОГИКИ КЛИНИ И ТРЕХЭЛЕМЕНТНЫЕ ЦЕПИ

Дмитрий Буй, Елена Шишацкая

***Аннотация**: Рассмотрена сильная и слабая трехзначные логики Клини. Показано возникновение сильной логики из обычной булевой логики путем применения общезначимой конструкции распространения операций с элементов на множества элементов в терминах полного образа. Проиллюстрировано компактное задание операций обоих логик Клини трехэлементными цепями.*

***Ключевые слова**: сильная логика Клини, слабая логика Клини, цепь, полный образ.*

***ACM Classification Keywords**: F.4.1 Theory of Computation – Mathematical Logic and Formal Languages - Mathematical Logic.*

***Conference**: The paper is selected from XIV[th] International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Работа посвящена сильной и слабой трехзначным логикам Клини, использующихся в теории рекурсии [1, с. 296-303]. Сильная логика используется в системах алгоритмических алгебр Глушкова [2, с. 117; § 4.2, с. 127], современных SQL-подобных языках реляционных баз данных [3, с. 169-170] и современных языках спецификаций UML/OCL при работе с булевым типом, пополненным третьим специальным значением [4, 5]. Заметим, что слабая логика Клини возникает путем естественного расширения в понимании [6] стандартных булевых операций, этот подход полностью отвечает принципам работы со специальным значением (UNDEFINED) стандарта объектных баз данных ODMG, в частности, языку запросов OQL [7, 8]. Далее под сильной и слабой логикой Клини будем понимать сильную и слабую трехзначную логику Клини.

Построение сильной логики Клини на основе обычной булевой логики

Применим общезначимую конструкцию распространения операций с элементов на множества элементов в терминах полного образа; именно такая конструкция применялась в [3, с. 23-24] при исследовании операций табличных алгебр, построенных на основе известных реляционных алгебр Кодда; общим свойствам полного образа посвящена работа [9].

Полный образ позволяет естественно распространять унарные (бинарные) операции на универсуме на булеан универсума. Через $[f]$ обозначим унарную тотальную операцию на булеане $P(D)$ универсума D, которая индуцируется частичной операцией f на универсуме и задается равенством $[f](X) \stackrel{def}{=} f[X]$; тут и далее $f[X] \stackrel{def}{=} \{y \mid \exists x(x \in X \wedge y \simeq f(x))\}$ – полный образ множества X относительно операции f, где, учитывая частичность функции, $\simeq$ – обобщенное равенство. Аналогично, пусть F – бинарная частичная операция на D; она также порождает бинарную тотальную операцию $[F]$ на булеане универсума D, которая задается равенством $[F](X,Y) \stackrel{def}{=} F[X \times Y]$.

Применим указанную схему расширения к сигнатурным операциям алгебры стандартной логики $<\{T,F\};\wedge,\vee,\neg>$, где T,F – логические значения истины и лжи соответственно. Результат расширения операции конъюнкции $\wedge$ и отрицания $\neg$ на булеан $P(\{T,F\})$ приведены в таблицах 1, 2 (расширение дизъюнкции строится аналогично).

Таблица 1. Операция $[\wedge]$ на булеане $P(\{T,F\})$

	$\varnothing$	$\{T\}$	$\{F\}$	$\{T,F\}$
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$\{T\}$	$\varnothing$	$\{T\}$	$\{F\}$	$\{T,F\}$
$\{F\}$	$\varnothing$	$\{F\}$	$\{F\}$	$\{F\}$
$\{T,F\}$	$\varnothing$	$\{T,F\}$	$\{F\}$	$\{T,F\}$

Таблица 2. Операция $[\neg]$ на булеане $P(\{T,F\})$

аргумент	$\varnothing$	$\{T\}$	$\{F\}$	$\{T,F\}$
значение	$\varnothing$	$\{F\}$	$\{T\}$	$\{T,F\}$

В таблице 1 первому аргументу отвечают столбцы, второму аргументу – строки. Расширения бинарных операций коммутативны; поэтому таблица 1 “симметрична” (относительно главной диагонали), и сопоставление аргументам столбцов или строк в действительности несущественно. Свойства коммутативности и ассоциативности расширений конъюнкции и дизъюнкции наследуются (что следует из общих результатов [3, утверждение 1.3.1; 9, утверждение 5]). Поскольку декартово произведение и полный образ сохраняют пустое множество, то и операции $[\wedge]$, $[\vee]$, $[\neg]$ сохраняют пустое множество. Поэтому в таблице 1, например, присутствуют константные строка и столбец, заполненные $\varnothing$.

Рассмотрим отображение $\psi : \{T,F,\omega\} \to P(\{T,F\})$, где ω – третье логическое значение логики Клини (содержательно интерпретируется как неопределенность): $\psi(T) = \{T\}$, $\psi(F) = \{F\}$, $\psi(\omega) = \{T,F\}$. Очевидно, отображение инъективно, но не сюръективно (ведь пустое множество не входит в область значений отображения ψ). Операции алгебры сильной логики Клини будем обозначать как операции алгебры стандартной логики, вводя только нижний индекс k; договоримся об одноименных операциях: операциям $\wedge_k$, $\vee_k$ и $\neg_k$ сопоставляются соответственно операции $[\wedge]$, $[\vee]$ и $[\neg]$.

Предложение 1 (построение алгебры сильной логики Клини). Отображение ψ – однозначный гомоморфизм алгебры сильной логики Клини $< \{T,F,\omega\}; \wedge_k, \vee_k, \neg_k >$ в алгебру $< P(\{T,F\}); [\wedge],[\vee],[\neg] >$, то есть это отображение является вложением алгебры сильной логики Клини в алгебру $< P(\{T,F\}); [\wedge],[\vee],[\neg] >$. □

Доказательство. Действительно, заменяя в таблицах 1-2 значения $\{T\},\{F\},\{T,F\}$ на T,F,ω соответственно (согласно отображения ψ) и удаляя константные столбец и строку, заполненные значением $\varnothing$, приходим к табличному заданию операций конъюнкции и отрицания алгебры сильной логики Клини. Случай сильной дизъюнкции рассматривается полностью аналогично. □

Таким образом, алгебру сильной логики Клини можно получить путем применения к алгебре классической булевой логики конструкции расширения (в терминах полного образа) ее сигнатурных операций.

Компактное задание операций сильной логики Клини

Идея заключается в переходе от алгебры $< \{T,F,\omega\}; \wedge_k, \vee_k, \neg_k >$ к соответствующей структуре (отметим, что сейчас чаще употребляется термин “решетка”, однако будем пользоваться термином “структура”, поскольку ссылаемся на результаты [10], где используется именно этот термин).

Действительно, непосредственно проверяется, что эти сигнатурные операции коммутативны, ассоциативны и идемпотентны; кроме того, выполняются два закона поглощения: $x \vee_k (x \wedge_k y) = x$ и $x \wedge_k (x \vee_k y) = x$ для всех $x, y \in \{T,F,\omega\}$. Следовательно, по стандартной процедуре, положив $x \leq_k y \overset{def}{\Leftrightarrow} x = x \wedge_k y$ (эквивалентно $x \leq_k y \overset{def}{\Leftrightarrow} y = x \vee_k y$), можно перейти к структуре, точные грани двухэлементных множеств которой находятся по формулам: $\mathbf{inf}\{x,y\} = x \wedge_k y$, $\mathbf{sup}\{x,y\} = x \vee_k y$ [10, теорема 3, с. 154]. Отношение $\leq_k$ в общем случае является частичным порядком [10, теорема 1, с. 151-152]. Для алгебры сильной логики Клини оно проиллюстрировано в таблице 3 (значениям аргумента x

отвечают строки, y – столбцы; в следующих таблицах будем придерживаться этого же соглашения); знак "+" в ячейке означает, что соответствующие элементы находятся в отношении, знак "-" – не находятся.

Таблица 3. Порядок $\leq_k$ на $\{T,F,\omega\}$

	$x = x \wedge_k y$	$y = x \vee_k y$	$x = x \wedge_k y$	$y = x \vee_k y$	$x = x \wedge_k y$	$y = x \vee_k y$
x \ y	T		F		ω	
T	+	+	-	-	-	-
	+		-		-	
F	+	+	+	+	+	+
	+		+		+	
ω	+	+	-	-	+	+
	+		-		+	

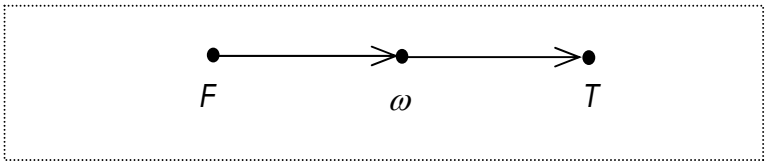


Рис.1. Линейный порядок $\leq_k$ на множестве $\{T,F,\omega\}$

Для случая структуры, отвечающей алгебре сильной логики Клини, ее порядок является линейным (см. рис. 1, построенный на основе таблицы 3), а именно $F \leq_k \omega \leq_k T$ (для компактности на рис. 1 не приведена одна стрелка, возникающая ввиду транзитивности, и три петли, отвечающие рефлексивности порядка). Таким образом, общая ситуация существенно упрощается: структура в действительности является цепью и $x \wedge_k y$ является наименьшим ($x \vee_k y$ – наибольшим) из элементов x, y для всех $x, y \in \{T,F,\omega\}$. Анализ процедуры построения структуры показывает, что линейность в общем случае частичного порядка обеспечивается таким свойством сильных конъюнкции и дизъюнкции – $x \wedge_k y, x \vee_k y \in \{x,y\}$ для всех $x, y \in \{T,F,\omega\}$.

Сформулируем общий результат для коммутативных идемпотентных полугрупп. В формулировке следующего утверждения считаем известной связь между коммутативными идемпотентными полугруппами и нижними полуструктурами [10, теорема 1, с. 151-152].

Предложение 2 (критерий линейности порядка полуструктуры, построенной по коммутативной идемпотентной полугруппе). Пусть $<D,+>$ – коммутативная идемпотентная полугруппа, а $\leq$ – частичный порядок, соответствующей нижней полуструктуре, то есть $x \leq y \overset{def}{\Leftrightarrow} x = x + y$. Порядок $\leq$ линейный (т.е. $<D,\leq>$ является цепью) тогда и только тогда, когда $x + y \in \{x,y\}$ для всех $x, y \in D$. □

Доказательство. Необходимость. Пусть порядок $\leq$ линеен, установим принадлежность $x + y \in \{x,y\}$ для всех $x, y \in D$. Пусть x, y – произвольные элементы; поскольку порядок линеен, то $x \leq y$ или наоборот $y \leq x$. В первом случае $x = x + y$, во втором – $y = y + x$. Поскольку операция коммутативна, то в обоих случаях выполняется принадлежность $x + y \in \{x,y\}$.

Достаточность. Пусть $x+y\in\{x,y\}$ для всех x,y; покажем, что порядок линеен. Пусть x,y – произвольные элементы, тогда по предположению $x+y=x$ или $x+y=y$. В первом случае $x\le y$ по определению порядка, во втором – $y\le x$ (действительно $y=x+y=y+x$). □

Из этого предложения и следует линейность порядка структуры, ассоциируемой с алгеброй сильной логики Клини. Кроме того, то, что структура $<\{T,F,\omega\};\le_k>$ является цепью, можно показать и другим элементарным путем. Действительно, указанная структура конечна, значит, она имеет наименьший (нуль) и наибольший (единицу) элементы, которые обозначим 0_k, 1_k соответственно. Поскольку структура трехэлементная, то, очевидно, что $0_k<_k 1_k$ и для третьего элемента z (отличного от наименьшего и наибольшего элементов) выполняется строгое неравенство $0_k<_k z<_k 1_k$. Следовательно, структура $<\{T,F,\omega\};\le_k>$ является цепью. Таким образом, линейность порядка структуры, ассоциируемой с алгеброй сильной логики Клини, следует, с одной стороны, из свойств операций (из принадлежностей $x\wedge_k y, x\vee_k y\in\{x,y\}$), а, с другой стороны, просто из трехэлементности структуры. Именно трехэлементность здесь существенна, ибо каждая n-элементная структура будет цепью при $n=1,2,3$, что не выполняется, в общем случае, при $n\ge 4$ (самый простой пример – булеан двухэлементного множества со стандартным порядком $\subseteq$). Подытожим вышеприведенную информацию.

Таблица 4. Операция $\wedge_\omega$ на $\{T,F,\omega\}$, сохраняющая ω

$\wedge_\omega$	T	F	ω
T	T	F	ω
F	F	F	ω
ω	ω	ω	ω

Предложение 3 (компактное задание бинарных операций алгебры сильной логики Клини). Отношение $\le_k$ превращает множество $\{T,F,\omega\}$ в цепь (а, значит, и в структуру), причем $x\wedge_k y$ является наименьшим (соответственно, $x\vee_k y$ – наибольшим) из элементов x,y для всех $x,y\in\{T,F,\omega\}$ согласно этого (линейного) порядка. □

Доказательство вытекает из общих результатов теории структур (интерпретации структуры как алгебры, каждая из двух сигнатурных операций которой идемпотентна, коммутативна и ассоциативна, а сами эти операции связаны законами поглощения) и линейности соответствующего порядка (см. рис. 1). □

Симптоматично, что по сути такое компактное задание операций (алгебры) сильной логики Клини используется в популярной программистской литературе по языку SQL: F интерпретируется как число 0, T – как 1, ω – как $\frac{1}{2}$; тогда $x\wedge_k y=\mathbf{min(x,y)}$, $x\vee_k y=\mathbf{max(x,y)}$ при естественном порядке – $0<\frac{1}{2}<1$; более того $\neg x=1-x$ (см., например, [11]).

Слабая логика Клини, возникающая при естественном расширении булевой логики

Рассмотрим слабую логику Клини и начнем с алгебры (группоида) $<\mathbf{\{T,F,\omega\};}\wedge_\omega>$, где операция конъюнкции отлична от рассмотренной выше сильной конъюнкции Клини и задана следующим образом: в случаях, когда хотя бы один аргумент равен третьему (новому) значению ω, результатом будет именно это значение; во всех других случаях операция ведет себя как операция $\wedge$ стандартной логики (таблица 4). Такую операцию $\wedge_\omega$ будем называть слабой конъюнкцией Клини [1].

Следовательно, речь идет о расширении стандартной конъюнкции, сохраняющем третье логическое значение. Операции конъюнкции и дизъюнкции алгебры сильной логики Клини, в отличие от операции отрицания этой же алгебры, третье логическое значение ω не сохраняют (см. таблицы 1, 2; например,

$T \vee \omega = T$, $F \wedge \omega = F$, но $\neg\omega = \omega$). Так определенная операция ассоциативна, коммутативна и идемпотентна (что проверяется непосредственно); то есть $<\{T,F,\omega\};\wedge_\omega>$ – коммутативная идемпотентная полугруппа и можно применить общую процедуру построения по ней полуструктуры (верхней или нижней).

Определим два бинарных отношения $x \le (\wedge_\omega) y \overset{def}{\Leftrightarrow} x = x \wedge_\omega y$ и $x \preceq (\wedge_\omega) y \overset{def}{\Leftrightarrow} y = x \wedge_\omega y$. Будем использовать обозначение вида $\le(\wedge_\omega)$, желая подчеркнуть, что отношение $\le$ индуцируется операцией $\wedge_\omega$, аналогично, для инверсного отношения. Иногда в таких обозначениях операцию явно указывать не будем. Тогда каждое из этих отношений является порядком, и множество $\{T,F,\omega\}$ с порядком $\le(\wedge_\omega)$ (с порядком $\preceq(\wedge_\omega)$) является нижней (верхней) полуструктурой, причем $\mathbf{inf}_{\le}\{x,y\} = x \wedge_\omega y$ (соответственно $\mathbf{sup}_{\preceq}\{x,y\} = x \wedge_\omega y$) [10, с. 152, теорема 1]. Очевидно, порядки $\le(\wedge_\omega)$ и $\preceq(\wedge_\omega)$ взаимоинверсны, то есть $x \le y \Leftrightarrow y \preceq x$. Следовательно, согласно принципа двойственности, не суть важно, какой именно порядок из этих двух рассматривать [12, с. 10]).

Таблица 6. Операция $\vee_\omega$ на $\{T,F,\omega\}$, сохраняющая ω

$\vee_\omega$	T	F	ω
T	T	T	ω
F	T	F	ω
ω	ω	ω	ω

Порядки $\le(\wedge_\omega)$ и $\preceq(\wedge_\omega)$ проиллюстрированы в таблице 5 и на рис. 2.

Таблица 5. Порядки $\le(\wedge_\omega)$ и $\preceq(\wedge_\omega)$ на $\{T,F,\omega\}$

	$x \le (\wedge_\omega) y \overset{def}{\Leftrightarrow} x = x \wedge_\omega y$			$x \preceq (\wedge_\omega) y \overset{def}{\Leftrightarrow} y = x \wedge_\omega y$		
x \ y	T	F	ω	T	F	ω
T	+	-	-	+	+	+
F	+	+	-	-	+	+
ω	+	+	+	-	-	+

Рис. 2. Порядки $\le(\wedge_\omega)$ (слева) и $\preceq(\wedge_\omega)$ (справа) на $\{T,F,\omega\}$

Аналогично, группоид $<\{T,F,\omega\};\vee_\omega>$, операция которого есть расширением стандартной операции дизъюнкции и сохраняет третье логическое значение ω (таблица 6), является коммутативной идемпотентной полугруппой. Снова можно применить общую процедуру построения полструктуры.

Определим два отношения $x \le (\vee_\omega) y \overset{def}{\Leftrightarrow} x = x \vee_\omega y$ и $x \preceq (\vee_\omega) y \overset{def}{\Leftrightarrow} y = x \vee_\omega y$ (как и для случая конъюнкции эти два отношения взаимоинверсны). Тогда каждое из этих отношений есть порядком, и множество $\{T,F,\omega\}$ с порядком $\le(\vee_\omega)$ ($\preceq(\vee_\omega)$) является нижней (верхней) полуструктурой, причем

$\mathbf{inf}_{\leq}\{x,y\} = x \vee_{\omega} y$ (соответственно $\mathbf{sup}_{\preceq}\{x,y\} = x \vee_{\omega} y$) [10, с. 151, теорема 1]. Порядки проиллюстрированы в таблице 7 та на рис. 3.

Таблица 7. Порядки $\leq(\vee_{\omega})$ и $\preceq(\vee_{\omega})$ на $\{T,F,\omega\}$

	$x \leq (\vee_{\omega}) y \Leftrightarrow x = x \vee_{\omega} y$			$x \preceq (\vee_{\omega}) y \Leftrightarrow y = x \vee_{\omega} y$		
x \ y	T	F	ω	T	F	ω
T	+	+	-	+	-	+
F	-	+	-	+	+	+
ω	+	+	+	-	-	+

Рис. 3. Порядки $\leq(\vee_{\omega})$ (слева) и $\preceq(\vee_{\omega})$ (справа) на $\{T,F,\omega\}$

Анализ процедуры построения полуструктур (напомним, что, например, $x \leq (\wedge_{\omega}) y \Leftrightarrow x = x \wedge_{\omega} y$), показывает, что линейность, в общем случае, частичного порядка обеспечивается таким свойством слабой конъюнкции и слабой дизъюнкции – $x \wedge_{\omega} y, x \vee_{\omega} y \in \{x,y\}$ для всех $x,y \in \{T,F,\omega\}$. Следовательно, ситуация с линейностью порядков аналогична сильной логике Клини и даже упрощается: принадлежности $x \wedge_{\omega} y$, $x \vee_{\omega} y \in \{x,y\}$ в случае, когда хотя бы один из аргументов x, y есть ω, автоматически следуют из сохранения значения ω операциями. Отличие заключается в том, что установить линейность порядка как следствие трехэлементности нельзя, поскольку работаем в полуструктурах (заметим, что существует простой пример трехэлементной полуструктуры, которая не является цепью; вместе с тем понятно, что двухэлементные полуструктуры являются цепями).

Подытожим информацию о полуструктурах (в действительности, структурах, поскольку порядок линеен), индуцированных двумя рассмотренными коммутативными идемпотентными полугруппами.

Предложение 4 (структуры, индуцируемые слабой конъюнкцией и слабой дизъюнкцией). Отношения $\leq(\wedge_{\omega})$ и $\preceq(\wedge_{\omega})$ превращают множество $\{T,F,\omega\}$ в цепь (а, значит, и в структуру), причем $x \wedge_{\omega} y$ является наименьшим (наибольшим) из элементов x,y согласно порядка $\leq(\wedge_{\omega})$ (соответственно $\preceq(\wedge_{\omega})$) для всех $x,y \in \{T,F,\omega\}$. Отношения $\leq(\vee_{\omega})$ и $\preceq(\vee_{\omega})$ также превращают множество $\{T,F,\omega\}$ в цепь (а, значит, и в структуру), причем $x \vee_{\omega} y$ является наименьшим (наибольшим) из элементов x,y согласно порядка $\leq(\vee_{\omega})$ (соответственно $\preceq(\vee_{\omega})$) для всех $x,y \in \{T,F,\omega\}$. □

Доказательство следует из указанных результатов теории полуструктур (взгляда на полуструктуру как на коммутативную идемпотентную полугруппу) и линейности соответствующих порядков (см. рис. 2-3). □

Очевидно, что в последнем предложении два (взаимоинверсные) порядка для слабой конъюнкции отличаются от двух (взаимоинверсных) порядков для слабой дизъюнкции. Следовательно, среди указанных четырех порядков нет порядка, отвечающего одновременно слабой конъюнкции и слабой дизъюнкции (в отличие от сильной логики Клини, где такой "общий порядок" существует, см. рис. 1). Для алгебры $<\{T,F,\omega\}; \wedge_{\omega}, \vee_{\omega}>$ сформулируем общий вопрос: существует ли на множестве $\{T,F,\omega\}$ порядок (обозначим его $\lhd$), превращающий его в структуру, причем для произвольных x,y выполняются

равенства $x \wedge_\omega y = \mathbf{inf}_\lhd \{x, y\}$, $x \vee_\omega y = \mathbf{sup}_\lhd \{x, y\}$ (либо, согласно принципа двойственности, в эквивалентной форме $x \wedge_\omega y = \mathbf{sup}_\lhd \{x, y\}$, $x \vee_\omega y = \mathbf{inf}_\lhd \{x, y\}$)? Допустим, что такой порядок существует, тогда должны выполняться законы поглощения для операций слабой конъюнкции и слабой дизъюнкции [10, с. 152-153, теорема 2]. Но результаты непосредственной проверки этих законов, приведенные в таблице 8, показывают, что в случаях, когда только y совпадает с ω, оба закона поглощения не выполняются. Следовательно, такого порядка не существует.

Таблица 8. Выполнимость законов поглощения для операций $\vee_\omega, \wedge_\omega$

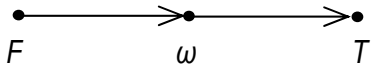
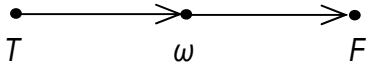
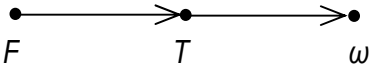
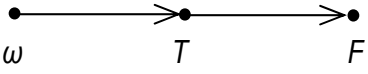
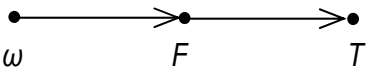
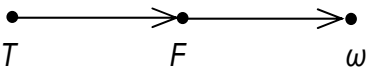
	$(x \vee_\omega y) \wedge_\omega x = x$			$(x \wedge_\omega y) \vee_\omega x = x$		
x \ y	T	F	ω	T	F	ω
T	+	+	-	+	+	-
F	+	+	-	+	+	-
ω	+	+	+	+	+	+

Невыполнение законов поглощения вполне естественно. Ведь суть этих законов заключается в том, что значения выражений, $(x \vee_\omega y) \wedge_\omega x$, $(x \wedge_\omega y) \vee_\omega x$ не зависят от значения y, а определяются только значением x. Понятно, что это требование не выполняется, если операции сохраняют значение ω, y совпадает с ним, а x, наоборот, отличный от ω.

Заключение

На множестве $\{T, F, \omega\}$ существует 6=3! возможных линейных порядков, связь которых с операциями дизъюнкции и конъюнкции двух рассмотренных логик Клини приведена в таблице 9.

Таблица 9. Всевозможные цепи на $\{T, F, \omega\}$ и их связь с логиками Клини

1.  $x \wedge_k y = \mathbf{min}(x, y)$ $x \vee_k y = \mathbf{max}(x, y)$	2.  $x \wedge_k y = \mathbf{max}(x, y)$ $x \vee_k y = \mathbf{min}(x, y)$
3.  $x \vee_\omega y = \mathbf{max}(x, y)$	4.  $x \vee_\omega y = \mathbf{min}(x, y)$
5.  $x \wedge_\omega y = \mathbf{min}(x, y)$	6.  $x \wedge_\omega y = \mathbf{max}(x, y)$

Порядок 1 (инверсный ему порядок 2) отвечает одновременно дизъюнкции и конъюнкции сильной логики Клини. Это порядки структуры, ассоциируемой с алгеброй сильной логики Клини. Порядок 3 (инверсный ему порядок 4) отвечает дизъюнкции, но не конъюнкции слабой логики Клини. Дуально, порядок 5 (инверсный ему порядок 6) – отвечает конъюнкции, но не дизъюнкции слабой логики Клини. Это порядки полуструктур, ассоциируемых с двумя полугруппами слабой логики Клини (сигнатура одной полугруппы состоит из слабой дизъюнкции, сигнатура другой – из слабой конъюнкции).

Литература

[1] Клини С.К. Введение в метаматематику. – Москва: ИЛ, 1957. – 526 с.

[2] Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – К.: Наукова думка, 1978. – 318 с.

[3] Редько В.Н., Брона Ю.Й., Буй Д.Б., Поляков С.А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. – К.: Видавничий дім “Академперіодика”, 2001. – 198 с.

[4] Cook S., Kleppe A., Mitchell R., Rumpe B., Warmer J., Wills A. The Amsterdam Manifesto on OCL. – UML 2.0 Request for information response: OMG Analysis & Design PTF, 1999 / http://www.trireme.com/whitepapers/design/components/OCL_manifesto.PDF.

[5] www.omg.org // 05-06-06.pdf.

[6] Манна З. Теория неподвижной точки программ // Кибернетический сборник. Вып. 15. – М.: Мир, 1978. – С. 38-100.

[7] The Object Data Standard: ODMG 3.0/ Edited by R.G.G. Cattel, Douglas K.Barry. – Morgan Kauffmann Publishers, 2000.

[8] http://www.omg.org/docs/omg/04-07-02.pdf.

[9] Буй Д.Б., Кахута Н.Д. Властивості теоретико-множинних конструкцій повного образу та обмеження // Вісник Київського університету. Сер.: фіз.-мат. науки. – 2005. – Вип. 2. – С. 232-240.

[10] Скорняков Л.А. Элементы алгебры. – Москва: Наука, 1986. – 240°с.

[11] http://www.sql-ex.ru/help/select2.php.

[12] Скорняков Л.А. Элементы теории структур. – Москва: Наука, 1982. – 160 с.

Информация об авторах

Буй Дмитрий Борисович *– заведующий лабораторией проблем программирования, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: buy@unicyb.kiev.ua*

Шишацкая Елена Владимировна *– инженер-программист лаборатории проблем программирования, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: shyshatskaja@unicyb.kiev.ua*

АВТОМАТНОЕ ПРЕДСТАВЛЕНИЕ ОНТОЛОГИЙ И ОПЕРАЦИИ НА ОНТОЛОГИЯХ

Сергей Крывый, Александр Ходзинский

Аннотация. *Предлагается подход к представлению онтологий в виде конечного автомата. Такое представление позволяет ввести операции на онтологиях, используя операции на регулярных языках. Операции на онтологиях дают возможность автоматизировать процесс анализа и синтеза онтологий и их составляющих частей.*

Ключевые слова*: онтологии, операции, конечные автоматы*

ACM Classification Keywords*: H4m. Miscellaneous*

Conference*: The paper is selected from XIV[th] International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

В последнее время в естественных науках и, в частности, в теоретическом программировании появилось столько различных направлений, течений и теоретических результатов, что становится проблематичным охватить хотя бы малую часть поля научной деятельности даже в отдельно взятых областях. Одним из подходов к пониманию взаимосвязей между различными течениями и теориями является онтологический подход [1,2]. Кроме того, в связи с возрастанием сложности программного и технического обеспечения вычислительных процессов требуется интеллектуализация этих процессов и такой интеллектуализации можно достичь, по мнению многих специалистов, путем использования онтолого-управляемых систем поиска, извлечения и обработки знаний, содержащихся в онтологиях. Онтологический подход для построения связей между понятиями некоторой предметной области, как правило, основывается на определении отношения *«предметная область – свойства - модели – приложения»*. В данной работе рассматривается способ представления онтологий с помощью конечных автоматов с одной стороны, и отношений, лежащих в основе каждой онтологии. Этот подход позволяет ввести операции на онтологиях используя операции на языках и автоматах. При таком подходе типы онтологий и их иерархия не детализируется с целью подчеркивания общности рассматриваемых операций. Операции иллюстрируются на простых примерах онтологий, относящихся к компьютерной математике [3].

Автоматное представление онтологий

Будем предполагать, что онтологии представляются в виде орграфа $G = (V, E)$, где множество вершин V представляет множество предметных областей, а множество ребер E – бинарное отношение между этими предметными областями. С каждым таким орграфом $G = (V, E)$ будем ассоциировать конечный (вообще говоря) частичный детерминированный автомат без выходов $A = (V, X=V, f, S, F)$, где V – множество состояний, которое также служит входным алфавитом данного автомата, S – подмножество начальных состояний, F – подмножество заключительных состояний (которое, в частности, может быть пустым), а функция переходов данного автомата определяется следующим образом: $f(u,v) = v$ тогда и только тогда, когда $(u,v) \in E$ и не определено в остальных случаях.

Рассмотрим пример представления фрагмента онтологии для предметной области «Комбинаторика».

Пример 1. Пусть задана онтология, отражающая малую часть предметной области «Комбинаторика», в виде следующего орграфа:

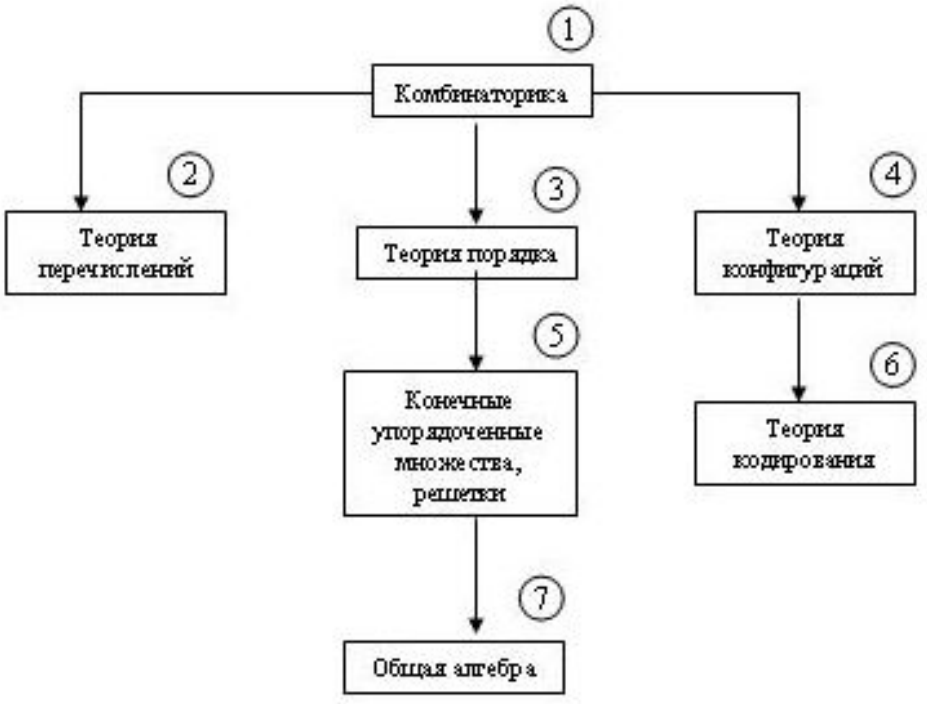


Рис. 1. Онтология O

Соответствующий данной онтологии конечный автомат имеет вид *A = (V = {1,2, 3, 4,5, 6,7}, X = {1,2,3,4,5,6,7}, f, {1}, {7})*, где *f* задана таким графом переходов:

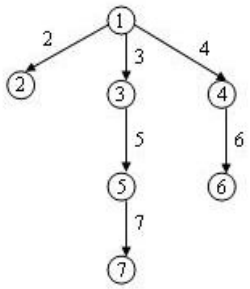


Рис. 2. Конечный автомат A для O

Это значит, что *f(1,2) = 2, f(1,3) =3, f(1,4) =4, f(3,5) =5, f(5,7) =7, f(4,6) =6.* Остальные переходы в данном автомате неопределенны.

Операции на онтологиях в автоматном представлении

Представление онтологий в виде конечного автомата без выходов позволяет ввести операции на онтологиях. Операции на автоматах означают операции на регулярных языках, которые акцептируются этими автоматами. Основными такими операциями являются следующие:

- ***объединение –*** теоретико-множественное объединение множества состояний и множества переходов данных автоматов-аргументов;
- ***пересечение*** – теоретико-множественное пересечение множества состояний и множества переходов, пополненное транзитивным замыканием отношения достижимости на автоматах-аргументах;
- ***конкатенация*** или ***умножение*** двух автоматов – частный случай операции объединения, когда объединение выполняется только по множеству начальных состояний второго автомата;

- ***итерация*** – повторяемая конечное число раз операция умножения, применяемая в рамках одной онтологии с целью уточнения и пополнения этой онтологии (эта операция практически означает пошаговое уточнение и пополнение онтологий);
- ***обращение*** – ориентация в противоположном направлении переходов в автомате, представляющем данную онтологию, т. е. построение функции переходов $g(v,u) = u$ тогда и только тогда, когда $f(u,v) = v$ и неопределенно в остальных случаях.

Пример 2. Пусть дана онтология вида

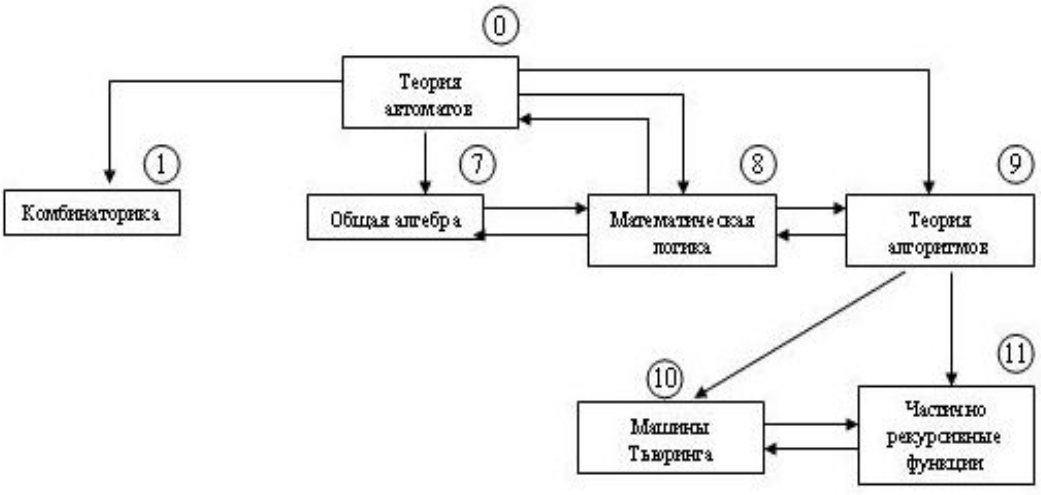


Рис. 3. Онтология O_1

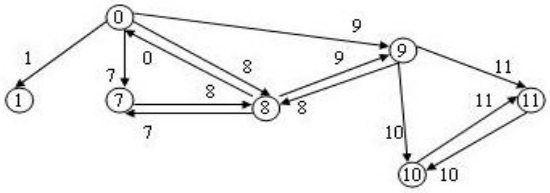


Рис. 4. Автомат A_1 для онтологии O_1

где $A_1=(\{0,1,7,8,9,10,11\},\{0,1,\ldots,11\},g,\{0\},\{11\})$.

Тогда введенные выше операции дают такие результаты, если их применить к автоматам A_1 и A из предыдущего примера.

Объединение:

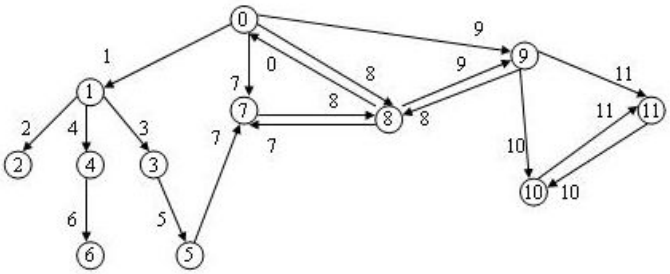


Рис. 5. Автомат $A \cup A_1$

Пересечение:

Рис. 6. Автомат $A \cap A_1$

Итерация: уточнение онтологии O_1:

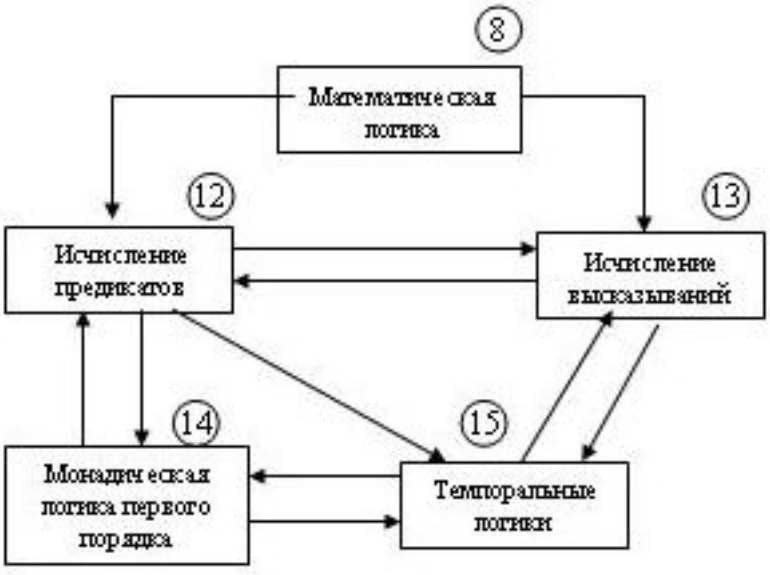


Рис. 7. Уточнение O_2 для онтологии O_1

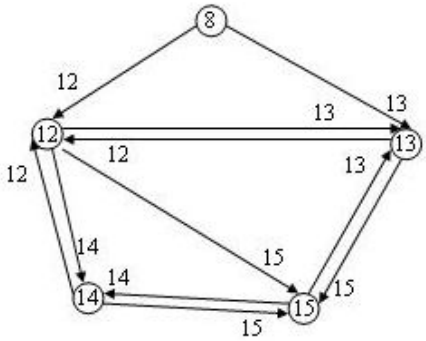


Рис. 8. Автомат A_2 для онтологии O_2

Конкатенируя автоматы A_1 и A_2 по начальному состоянию 8 автомата A_2, получаем автомат, представляющий уточненную онтологию $O_1 * O_2$.

Обращение: применяя эту операцию к A_1, получаем автомат:

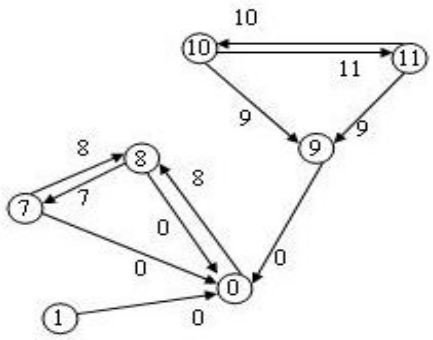


Рис. 9. Автомат обращения для онтологии O_1

Краткая характеристика операций

Алгебраические свойства введенных операций на онтологиях вытекают из соответствующих свойств операций алгебры регулярных языков. Это значит, что данные операции удовлетворяют следующим законам: коммутативность и ассоциативность операций объединения и пересечения, ассоциативность умножения, дистрибутивность операции умножения относительно операций объединения и пересечения.

Данное множество операций (в случае надобности) можно расширять по крайней мере в двух направлениях. Одним из таких направлений является расширение операциями на графах (введение и удаление вершины и ребра, соединение графов, изоморфного соединения [6], декартового произведения и т. д.). Другим направлением является алгебра отношений. Поскольку каждая онтология является представлением некоторой совокупности отношений (в частности: одного), то можно вводить операции реляционной алгебры.

Какое из возможных направлений будет выбрано, зависит от практических потребностей использования онтологий. Прогнозировать что-либо на этот счет не имеет смысла, так как практика оказывается всегда богаче любой теории. Авторы надеются, что представленные операции над онтологиями окажутся полезными при анализе, синтезе и манипулировании онтологиями и онтологическими объектами.

Проблемы реализации операций

Рассмотрим теперь некоторые проблемы, возникающие на пути реализации данных операций.

Первая проблема (и возможно основная при работе с онтологиями) связана с тем, что корректное выполнение описанных выше операций требует создания некоторого общего глоссария предметных областей и понятий, с помощью которого можно было бы однозначно идентифицировать соответствующие объекты. По видимому, эта проблема является не только проблемой на пути реализации введенных операций, но и в некотором смысле общей проблемой на пути построения онтологий и работы с онтологиями.

Вторая проблема, возникающая при реализации операций, связана с имеющейся иерархией областей и понятий. Дело в том, что в различных онтологиях одни и т же понятия и объекты могут находиться на разных уровнях иерархии и это необходимо учитывать при применении операций. В предлагаемом подходе эта проблема решается с помощью построения транзитивного замыкания отношения достижимости на состояниях автоматов, представляющих данные онтологии. Однако, авторы не уверены в том, что этого замыкания достаточно для решения проблемы. Здесь, по-видимому, необходимы эксперименты на реальных онтологиях и их представлениях.

Третья проблема связана с полнотой знаний, имеющихся в представленных онтологиях. Эта проблема является основной в процессе спецификации и верификации программного и технического обеспечения. Здесь же эта проблема связана с возможностью построения в некотором (хотя бы) смысле полной онтолого-управляемой информационной системы.

Заключение

Тема данной работы возникла в связи с докладами, которые были представлены на конференциях KDS-2005 и KDS-2007 (Варна, Болгария). Раздел по онтологиям на этих конференциях был одним из самых больших и доклады, представленные в этих разделах, были стимулирующими для разработки представления онтологий и операций на онтологиях с целью автоматизации процесса проектирования и манипулирования этими объектами. Возможно после данной попытки ввести операции на онтологиях появятся и другие подходы к построению алгебры онтологий, что было бы весьма желательным и плодотворным для развития этой области знаний. Наш подход, по видимому, не является самым лучшим, поскольку требует решения перечисленных выше проблем.

Библиография

[1] Gavrilova T., Puuronen S. In Search of a Vision: Ontological View on User Modeling Conferences' Scope. XII-th International Conference KDS 2007, ITHEA, Sofhia, 2007. Volume 2, p.422-427.

[2] Gribova V. Automatic Generation of Context-sensitive Help Using a User Interface Project. XII-th International Conference KDS 2007, ITHEA, Sofhia, 2007. Volume 2, p.417-422.

[3] Кривой С., Матвеева Л., Лукъянова Е., Седлецкая О. Онтологический взгляд на теорию автоматов. XII-th International Conference KDS 2007, ITHEA, Sofhia, 2007. Volume 2, p.427-436.

[4] Artemieva I. XII-th International Conference KDS 2007, ITHEA, Sofhia, 2007. Volume 2, p.403-411.

[5] Кривий С. Л. Дискретна математика. Вибрані питання. Київ. Видавничий дім „Києво-Могилянська академія". - 2007. - 572 с.

[6] Kryvyy S. , Hajder M., Dymora P., Mazurek M. Designing of the computer network topologies and coherent graphs algebra. - In Annales Universitatis Mariae Curie-Sklodowska, Sectia A1.- Informatica.-Poland.-Lublin. -2006. - v. 5. - p. 379-391.

Информация об авторах

Крывый Сергей – *Институт кибернетики им. В.М.Глушкова НАН Украины, Украина, Киев, 03187, Проспект Глушкова 40, Институт кибернетики, e-mail: krivoi@i.com.ua*

Ходзинсккий Александр – *Институт кибернетики им. В.М.Глушкова НАН Украины, Украина, Киев, 03187, Проспект Глушкова 40, Институт кибернетики, e-mail: ho@cyber.kiev.ua*

МЕРА ОПРОВЕРЖИМОСТИ ВЫСКАЗЫВАНИЙ ЭКСПЕРТОВ, РАССТОЯНИЯ В МНОГОЗНАЧНОЙ ЛОГИКЕ И ПРОЦЕССЫ АДАПТАЦИИ

Александр Викентьев

***Аннотация**: В работе определяются и доказываются свойства расстояний на высказываниях экспертов в многозначной логике и изучается мера опровержимости таких высказываний. Получены обобщения на общий случай результатов, доказанных ранеее в случае 2-значного и 3-значного исчислений [1].*

***Ключевые слова**: многозначные экспертные высказывания, метрика на высказываниях (expert statements, metric).*

***ACM Classification Keywords**: I.2.6. Исскуственный интеллект, изучение баз знаний, процессы адаптации (Artificial Intelligence - knowledge acquisition).*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

В настоящее временя появляется все больший интерес к построению логических решающих функций на основе анализа экспертной информации, заданной в виде вероятностных логических высказываний нескольких экспертов, реализации адаптивных методов и согласования высказываний [1-8] .

В данной работе предложено записывать высказывания экспертов в виде формул многозначной логики Лукасевича [3]. При организации поиска логических закономерностей требуются как расстояния между высказываниями экспертов и формулами в моделях (по базе знаний) в произвольный текущий момент времени, так и мера опровержимости. Последняя позволяет ранжировать высказывания по степени их нетривильности, важности. Планируемая обработка сообщений экспертов в различные моменты (срезы) времени показывает, что поскольку гипотезы-предположения у экспертов вообще говоря меняются, то расстояния и мера опровержимости тоже могут изменяться. Значит происходит адаптация во времени как самой теории, так и расстояний между высказываниями и их опровержимостей. Аппарат для обработки таких знаний подготовлен в работах Викентьева А.А., начатых совместно с Лбовым Г.С. и Кореневой Л.Н., а согласования знаний- высказываний в работах Лбова-Герасимова [4-6] в классе логических решающих функций. Сигнал о смене класса моделей (а значит и теории) будет исходить либо от самих экспертов (по их изменяющимся знаниям) или при получении неправильных результатов при использовании старой (базы знаний) теории. В случае n=2, n=3 проведены теоретические исследования по указанным выше вопросам. Здесь рассмотрен случай для произвольного n . Конечно, не все ранее доказанные результаты для малых n переносятся на общую ситуацию.

Ясно, что различные высказывания экспертов (и соответствующие им формулы) несут в себе разное количество информации, а значит, возникает вопрос о ранжировании высказываний экспертов и сравнении их по информативности (мере опровержимости). Для решения этих задач в работе будут введены и найдены свойства расстояния между формулами. Подробно рассмотрена мера опровержимости рассматриваемых формул.

Работа выполнена при финансовой поддержке гранта РФФИ 07-01-00331а.

Определение расстояния между высказываниями экспертов

Определение 1.1. Множество элементарных высказываний $S^n(\varphi)$, используемых при написании формул многозначной логики φ, назовем *носителем формулы* φ**.**

Определение 1.2. Назовем *носителем совокупности знаний* $S^n(\Sigma)$, объединение носителей формул, входящих в Σ , т.е. $S^n(\Sigma) = \bigcup_{\varphi\in\Sigma} S^n(\varphi)$.

Определение 1.3. Назовем *множеством возможных значений носителя* совокупности знаний $Q^n(\Sigma) = \{\varphi_{\frac{k}{n-1}} \mid \varphi \in S(\Sigma), k = 1,.., n-1\}$.

Определение 1.4. *Моделью* М назовем любое подмножество $Q^n(\Sigma)$ такое, что М не содержит одновременно $\varphi_{\frac{k}{n-1}}$ и $\varphi_{\frac{l}{n-1}}$ $\forall k \neq l \quad \forall \varphi \in Q(\Sigma)$

Множество всех моделей будем обозначать $P^n(S(\Sigma))$. Для упрощения записи, верхний индекс у формул, означающий значность высказывания, будем опускать когда это не вызывает трудностей.

Лемма 1.1. (о числе моделей $P^n(S(\Sigma))$)

$| P(S(\Sigma)) |= n^{|S(\Sigma)|}$.

Доказательство: докажем утверждение по индукции.
Пусть $S(\Sigma) = \{A\}$; $| S(\Sigma) |= 1$. Тогда $P(S) = \{\{A\}, \{A_{\frac{n-2}{n-1}}\},..,\{A_{\frac{1}{n-1}}\}\{\}\}$.

$| P(S(\Sigma)) |= n$

Пусть верно для $| S(\Sigma) |= k-1$; $S(\Sigma) = \{A^1, A^2,.., A^{k-1}\}$; $P(S(\Sigma)) = n^{|S(\Sigma)|}$;
Докажем для $| S(\Sigma) |= k$, т.е. $S(\Sigma) = \{A^1, A^2,...A^k\}$.

$$P(S(\Sigma')) = P(S(\Sigma)) \cup \{M \cup \{A_1^k\} \mid M \in P(S(\Sigma))\} \cup \{M \cup \{A^k_{\frac{n-2}{n-1}}\} \mid M \in P(S(\Sigma)) \cup .. \cup \{M \cup \{A^k_{\frac{1}{n-1}}\} \mid M \in P(S(\Sigma))\}\}$$

Докажем это. Очевидно, что

$$P(S(\Sigma')) \supseteq P(S(\Sigma)) \cup \{M \cup \{A_1^k\} \mid M \in P(S(\Sigma))\} \cup \{M \cup \{A^k_{\frac{n-2}{n-1}}\} \mid M \in P(S(\Sigma)) \cup .. \cup \{M\{A^k_{\frac{1}{n-1}}\} \mid M \in P(S(\Sigma))\}\}$$

Докажем обратное включение.
Пусть $M \in P(S(\Sigma'))$. Тогда

если $A_l^k \in M$, где $l \in \{\frac{n-1}{n-1}, \frac{n-2}{n-1},..,0\}$, тогда $M \setminus A_l^k \in P(S(\Sigma))$

если $A_l^k \notin M$, то $M \in P(S(\Sigma))$
Следовательно,

$$P(S(\Sigma')) \subseteq P(S(\Sigma)) \cup \{M \cup \{A^k\} \mid M \in P(S(\Sigma))\} \cup \{M \cup \{A^k_{\frac{n-2}{n-1}}\} \mid M \in P(S(\Sigma)) \cup .. \cup \{M \cup \{A^k_{\frac{1}{n-1}}\} \mid M \in P(S(\Sigma))\}\}$$

Значит,

$$| P(S(\Sigma')) |=| P(S(\Sigma)) | + | P(S(\Sigma)) | +...+ | P(S(\Sigma)) |= n \mid P(S(\Sigma)) |= n * n^{|S(\Sigma)|} = n^{|S(\Sigma)|+1} = n^{|S(\Sigma')|}.$$

ч.т.д.

Определение 1.5. Элементарная формула A принимает на модели M значение $\frac{k}{n-1}$, k =1, .., $n-1$, если $A_{\frac{k}{n-1}} \in M$, т.е. $M \models A_{\frac{k}{n-1}} \Leftrightarrow A_{\frac{k}{n-1}} \in M$.

Определение 1.6. Элементарная формула A принимает на модели M значение 0, если $A_{\underline{k}} \notin M \ \forall k = 1,..,n-1$.

Далее, используя определенные выше формулы, полагаем:

3) $M \models (A \& B)_{\underline{k}} \Leftrightarrow (M \models A_{\underline{p}}$ и $M \models B_{\underline{q}}) \ \min(p,q) = k$

4) $M \models (A \vee B)_{\underline{k}} \Leftrightarrow (M \models A_{\underline{p}}$ и $M \models B_{\underline{q}}) \ \max(p,q) = k$

5) $M \models (\neg A)_{\underline{k}} \Leftrightarrow M \models A_{\underline{n-1-k}}$

Во всех остальных случаях формулы принимают значения 0.

Введем обозначения:

$$Mod_{S(\Sigma)}(A)_{\underline{k}} = \{M \mid M \in P(S(\Sigma)), M \models A_{\underline{k}}\}$$

$$Mod_{S(\Sigma)}(A_0) = \{M \mid M \in P(S(\Sigma)), M \not\models A_{\underline{k}}, \forall k = 1,..., n-1\}$$

Таким образом, любой формуле φ такой, что $S(\varphi) \subseteq S(\Sigma)$ соответствует совокупность $Mod_{S(\Sigma)}(\varphi)_{\underline{k}}$, $k = 1,..,n-1$ моделей из $P(S(\Sigma))$, на которых φ принимает значения $\frac{k}{n-1}$, $k = 1,..,n-1$ соответственно.

Сформулируем некоторые теоретико-модельные свойства.

Лемма 1.2.

1) $Mod_{S(\Sigma)}((A \& B)_{\underline{k}}) = \bigcup_{p=k}^{n-1}((Mod_{S(\Sigma)}(A)_{\underline{p}} \cap Mod_{S(\Sigma)}(B)_{\underline{k}}) \cup (Mod_{S(\Sigma)}(A)_{\underline{k}} \cap Mod_{S(\Sigma)}(B)_{\underline{p}})$;

2) $Mod_{S(\Sigma)}((A \vee B)_{\underline{k}}) = \bigcup_{p=0}^{k}((Mod_{S(\Sigma)}(A)_{\underline{p}} \cup Mod_{S(\Sigma)}(B)_{\underline{k}}) \cup (Mod_{S(\Sigma)}(A)_{\underline{k}} \cup Mod_{S(\Sigma)}(B)_{\underline{p}})$;

3) $Mod_{S(\Sigma)}(\neg A)_{\underline{k}} = Mod_{S(\Sigma)}(A)_{\underline{n-1-k}}$;

4) $\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(A)_{\underline{k}} = P(S(\Sigma)) / Mod_{S(\Sigma)}(\neg A)_1$

Определение 1.7. Назовем формулы φ и ψ *эквивалентными* (далее коротко $\varphi \equiv \psi$), если $\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\varphi)_{\underline{k}} = \bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\psi)_{\underline{k}}$, т.е. они имеют одно и тоже множество моделей в каждом значении истинности. Это отношение является *отношением эквивалентности*.

Определение 1.8. *Расстоянием* между формулами φ и ψ (такое, что $S(\varphi) \cup S(\psi) \subseteq S(\Sigma)$) на множестве $P(S(\Sigma))$ назовем величину (обобщающую нормированную симметрическую разность для многозначного случая, что является естественным обобщением расстояния для (классической) 2- и 3-значной логик):

$$\rho_{S(\Sigma)}(\varphi,\psi) = \frac{|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\varphi_{\underline{k}} \& \psi_0)| + |\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\varphi_0 \& \psi_{\underline{k}})|}{n^{|S(\Sigma)|}}$$

Свойства расстояния

Утверждение (Свойства расстояния $\rho_{S(\Sigma)}$)

Для любых формул φ,ψ таких, что $S(\varphi)\cup S(\psi)\subseteq S(\Sigma)$ справедливы утверждения:

1) $0\le \rho_{S(\Sigma)}(\varphi,\psi)\le 1$;

2) $\rho_{S(\Sigma)}(\varphi,\psi) = \rho_{S(\Sigma)}(\psi,\varphi)$;

3) $\rho_{S(\Sigma)}(\varphi,\psi) =0 \Leftrightarrow \varphi\equiv\psi$;

4) $\rho_{S(\Sigma)}(\varphi,\psi) =1 \Leftrightarrow \bigcup_{l=1}^{n-1}\bigcup_{k=1}^{n-1}(Mod(\varphi)_{\frac{k}{n-1}} \overline{\cup} Mod(\psi)_{\frac{l}{n-1}}) = P(S(\Sigma))$, где $\overline{\cup}$ - прямое объединение

5) $\rho_{S(\Sigma)}(\varphi,\psi)\le \rho_{S(\Sigma)}(\varphi,\chi) + \rho_{S(\Sigma)}(\chi,\psi)$

6) Если $\varphi^1\equiv\varphi^2$, то $\rho_{S(\Sigma)}(\varphi^1,\psi) = \rho_{S(\Sigma)}(\varphi^2,\psi)$;

Доказательство:

1) Очевидно, что $0<\rho_{S(\Sigma)}(\varphi,\psi)<1$, причем верхняя и нижняя границы достижимы. Приведем примеры формул, на которых они достигаются.

$\rho_{S(\Sigma)}(\varphi,\varphi)=0$; Пусть φ - формула, не принимающая значение $\frac{k}{n-1}$, $k=1,..,n-2$, тогда $\neg\varphi$- так же не принимает значение $\frac{k}{n-1}$, $k=1,..,n-2$. Тогда $\rho_{S(\Sigma)}(\varphi,\neg\varphi)=1$.

2) Данное свойство очевидно, из определения расстояния и симметричности операций.

3) Докажем прямую (слева направо) импликацию.

$$\rho_{S(\Sigma)}(\varphi,\psi)=0 \Leftrightarrow \sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\varphi)_{\frac{k}{n-1}}|+|Mod_{S(\Sigma)}(\psi)_{\frac{k}{n-1}}|)-2\sum_{p=1}^{n-1}\sum_{q=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{p}{n-1}}\,\&\,\psi_{\frac{q}{n-1}})|=0$$

$$\sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\varphi)_{\frac{k}{n-1}}|+|Mod_{S(\Sigma)}(\psi)_{\frac{k}{n-1}}|=2\sum_{p=1}^{n-1}\sum_{q=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{p}{n-1}}\,\&\,\psi_{\frac{q}{n-1}})| \qquad (1)$$

По определению:

$$\sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\varphi)_{\frac{k}{n-1}}|=\sum_{k=1}^{n-1}\sum_{s=0}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}}\,\&\,\psi_{\frac{s}{n-1}})|;$$

$$\sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\psi)_{\frac{k}{n-1}}|=\sum_{k=1}^{n-1}\sum_{s=0}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{s}{n-1}}\,\&\,\psi_{\frac{k}{n-1}})|;$$

$$\Rightarrow\sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\varphi)_{\frac{k}{n-1}}|+|Mod_{S(\Sigma)}(\psi)_{\frac{k}{n-1}}|)=$$

$$=2\sum_{k=1}^{n-1}\sum_{s=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}}\,\&\,\psi_{\frac{s}{n-1}})|+\sum_{k=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}}\,\&\,\psi_0)|+\sum_{k=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_0\,\&\,\psi_{\frac{k}{n-1}})| \qquad (2)$$

Если из (2) вычесть (1), то получается

$\sum_{k=1}^{n-1}(|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}}\,\&\,\psi_0)|+|Mod_{S(\Sigma)}(\varphi_0\,\&\,\psi_{\frac{k}{n-1}})|)=0$; откуда следуют импликации

$$\Rightarrow\sum_{k=1}^{n-1}|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}}\,\&\,\psi_0)|=0 \;\Rightarrow\; \bigcup_{k=1}^{n-1}Mod(\varphi)_{\frac{k}{n-1}}\subseteq\bigcup_{k=1}^{n-1}Mod(\psi)_{\frac{k}{n-1}} \qquad (3)$$

$$\sum_{k=1}^{n-1}|Mod_{s(\Sigma)}(\varphi_0\,\&\,\psi_{\frac{k}{n-1}})|=0 \Rightarrow \bigcup_{k=1}^{n-1}Mod(\varphi)_{\frac{k}{n-1}}\supseteq\bigcup_{k=1}^{n-1}Mod(\psi)_{\frac{k}{n-1}} \qquad (4)$$

$\Rightarrow$ Из (3), (4) получаем $\bigcup_{k=1}^{n-1} Mod(\varphi)_{\frac{k}{n-1}} = \bigcup_{k=1}^{n-} Mod_{s(\Sigma)}(\psi)_{\frac{k}{n-1}} \quad \Rightarrow \quad \varphi = \psi$

Докажем обратное: если $\varphi = \psi$, то $\rho(\varphi,\psi) = 0$.

По определению $\varphi \equiv \psi$ означает, что $\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\varphi)_{\frac{k}{n-1}} = \bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\psi)_{\frac{k}{n-1}}$

$$\bigcup_{k=1}^{n-1} Mod(\varphi)_{\frac{k}{n-1}} \subseteq \bigcup_{k=1}^{n-1} Mod(\psi)_{\frac{k}{n-1}} \Rightarrow Mod(\varphi_{\frac{k}{n-1}} \& \psi_0) = \varnothing$$

$$|Mod(\varphi_{\frac{k}{n-1}} \& \psi_0)| = 0 \ \forall k \Rightarrow |Mod(\varphi_0 \& \psi_{\frac{k}{n-1}})| = 0 \ \forall k$$

$$\Rightarrow \rho(\varphi,\psi) = \frac{|\bigcup_{k=1}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \psi_0)| + |\bigcup_{k=1}^{n-1} Mod(\psi_{\frac{k}{n-1}} \& \varphi)|}{n^{|S(\Sigma)|}} = 0.$$

4) $\rho(\varphi,\psi) = 1 \Leftrightarrow |Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \& \psi_0)| + |\bigcup_{k=1}^{n-1} Mod_{S(\Sigma)}(\varphi_0 \& \psi_{\frac{k}{n-1}})| = n^{|S(\Sigma)|}$ **(5)**

$$n^{|S(\Sigma)|} = |\bigcup_{k=1}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \psi_0)| + |\bigcup_{k=1}^{n-1} Mod(\psi_{\frac{k}{n-1}} \& \varphi_0)| + |\bigcup_{p=1}^{n-1}\bigcup_{q=1}^{n-1} Mod(\varphi_{\frac{p}{n-1}} \& \psi_{\frac{q}{n-1}})| + |Mod(\varphi_0 \& \psi_0)|$$

Из (5) и вычислений получаем, что:

$$\bigcup_{p=1}^{n-1}\bigcup_{q=1}^{n-1} Mod(\varphi_{\frac{p}{n-1}} \& \psi_{\frac{q}{n-1}})| + |Mod(\varphi_0 \& \psi_0)| = 0;$$

т.е. φ и ψ одновременно не принимает значение 0. Если φ принимает значение не 0, то ψ обязательно равно 0; т.е. $\bigcup_{l=1}^{n-1}\bigcup_{k=1}^{n-1}(Mod(\varphi)_{\frac{k}{n-1}} \cup Mod(\psi)_{\frac{l}{n-1}}) = P(S(\Sigma))$. Т.е. модели $Mod(\varphi)_{\frac{k}{n-1}} \ \forall k = \overline{1,n-1}$ и $Mod(\psi)_{\frac{l}{n-1}} \ \forall l = \overline{1,n-1}$ образуют пересекающие множества, такие что их объединение заполняет все наше пространство.

5) $\rho(\varphi,\chi) = \frac{|\bigcup_{k=1}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \chi_0)| + \bigcup_{k=1}^{n-1} Mod(\varphi_0 \& \chi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}}$

$$\rho(\varphi,\psi) = \frac{|\bigcup_{k=1}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \psi_0)| + |\bigcup_{k=1}^{n-1} Mod(\varphi_0 \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}}$$

$$\rho(\varphi,\chi) = \frac{|\bigcup_{k=1}^{n-1} Mod(\psi_{\frac{k}{n-1}} \& \chi_0)| + |\bigcup_{k=1}^{n-1} Mod(\psi_0 \& \chi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}}$$

$$Mod(\varphi_{\frac{k}{n-1}} \& \chi) = \bigcup_{l=0}^{n-1}(\varphi_{\frac{k}{n-1}} \chi_0 \psi_{\frac{l}{n-1}}) \Rightarrow$$

$$\Rightarrow \rho(\varphi,\chi)n^{|S(\Sigma)|} = |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \chi_0 \& \psi_{\frac{l}{n-1}})| + |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\varphi_0 \& \chi_{\frac{k}{n-1}} \& \psi_{\frac{l}{n-1}})|$$

$$\rho(\varphi,\psi)n^{|S(\Sigma)|} = |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\varphi_{\frac{k}{n-1}} \& \psi_0 \& \chi_{\frac{l}{n-1}})| + |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\varphi_0 \& \psi_{\frac{k}{n-1}} \& \chi_{\frac{l}{n-1}})|$$

$$\rho(\psi,\chi)n^{|S(\Sigma)|} = |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\psi_{\frac{k}{n-1}} \,\&\, \chi_0 \,\&\, \varphi_{\frac{l}{n-1}})| + |\bigcup_{k=1}^{n-1}\bigcup_{l=0}^{n-1} Mod(\psi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \varphi_{\frac{l}{n-1}})|$$

$$\rho(\varphi,\chi)n^{|S(\Sigma)|} = \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\varphi_{\frac{k}{n-1}} \,\&\, \chi_0 \,\&\, \psi_{\frac{l}{n-1}})| + \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\varphi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \psi_{\frac{l}{n-1}})| +$$

$$+ \sum_{k=1}^{n-1} |\, Mod(\varphi_{\frac{k}{n-1}} \,\&\, \chi_0 \,\&\, \psi_0)| + \sum_{k=1}^{n-1} |\, Mod(\varphi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \psi_0)|$$

$$\rho(\varphi,\psi)n^{|S(\Sigma)|} = \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\varphi_{\frac{k}{n-1}} \,\&\, \psi_0 \,\&\, \chi_{\frac{l}{n-1}})| + \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\varphi_0 \,\&\, \psi_{\frac{k}{n-1}} \,\&\, \chi_{\frac{l}{n-1}})| +$$

$$+ \sum_{k=1}^{n-1} |\, Mod(\varphi_{\frac{k}{n-1}} \,\&\, \psi_0 \,\&\, \chi_0)| + \sum_{k=1}^{n-1} |\, Mod(\varphi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \psi_0)|$$

$$\rho(\psi,\chi)n^{|S(\Sigma)|} = \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\psi_{\frac{k}{n-1}} \,\&\, \chi_0 \,\&\, \varphi_{\frac{l}{n-1}})| + \sum_{k=1}^{n-1}\sum_{l=1}^{n-1} |\, Mod(\psi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \varphi_{\frac{l}{n-1}})| +$$

$$+ \sum_{k=1}^{n-1} |\, Mod(\psi_{\frac{k}{n-1}} \,\&\, \chi_0 \,\&\, \varphi_0)| + \sum_{k=1}^{n-1} |\, Mod(\psi_0 \,\&\, \chi_{\frac{k}{n-1}} \,\&\, \varphi_0)|$$

$$\Rightarrow \rho(\varphi,\chi) \le \rho(\varphi,\psi) + \rho(\psi,\chi);$$

6) $\varphi^1 \equiv \varphi^2$ обозначает, что $\bigcup_{k=1}^{n-1} Mod(\varphi^1)_{\frac{k}{n-1}} \equiv \bigcup_{k=1}^{n-1} Mod(\varphi^2)_{\frac{k}{n-1}}$

$\Rightarrow \bigcup_{k=1}^{n-1} Mod(\varphi^1{}_{\frac{k}{n-1}} \,\&\, \psi_0) \equiv \bigcup_{k=1}^{n-1} Mod(\varphi^2{}_{\frac{k}{n-1}} \,\&\, \psi_0)$ очевидно.

ч.т.д.

Следующая лемма дает возможность упрощения вычисления расстояния в нашей ситуации.

Лемма (о локальности вычисления расстояния и поведении при расширении)

Для любого $S(\Sigma_0)$ т. что $S(\varphi) \cup S(\psi) \subseteq S(\Sigma_0)$ и любого $S(\Sigma_1)$, т. что $S(\Sigma_0) \subset S(\Sigma_1)$ имеет место равенство: $\rho_{S(\Sigma_0)}(\varphi,\psi) = \rho_{S(\Sigma_1)}(\varphi,\psi)$.

Доказательство

Рассмотрим $S(\Sigma_1) = S(\Sigma_0) \cup \{\chi\}, \quad \chi \notin S(\Sigma_0)$.

При этом $P(S(\Sigma_1)) = P(S(\Sigma_0)) \cup (\bigcup_{k=1}^{n-1} \{M \cup \{\chi_{\frac{k}{n-1}}\} \mid M \in P(S(\Sigma_0))\})$ и $|\, P(S(\Sigma_1))| = n\,|\, P(S(\Sigma_0))|$

Также $|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_1)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi)| = n\,|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_0)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_0)|$

т.о. $P(S(\Sigma_1)) = \underbrace{\overbrace{P(S(\Sigma_0))}^{1} \cup \bigcup_{k=1}^{n-1} \underbrace{\{M \cup \{\chi_{\frac{k}{n-1}}\} \mid M \in P(S(\Sigma_0))\}}_{1}}_{n-1}^{n} \quad \Rightarrow$

$$\Rightarrow \rho_{S(\Sigma_1)}(\varphi,\psi) = \frac{|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_1)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_0)| + |\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_1)}(\varphi_0 \,\&\, \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma_1)|}} =$$

$$= \frac{n\,|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_0)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_0)\,| + n\,|\bigcup_{k=1}^{n-1} Mod_{S(\Sigma_0)}(\varphi_0 \,\&\, \psi_{\frac{k}{n-1}})\,|}{n \cdot n^{|S(\Sigma_0)|}} = \rho_{S(\Sigma_0)}(\varphi,\psi)$$

Пусть теперь $|\, S(\Sigma_1) \setminus S(\Sigma_0)\,| = |\,\{A^1,...,A^m\}\,| = m \geq 1$.

Тогда $\rho_{S(\Sigma_0)}(\varphi,\psi) = \rho_{S(\Sigma_0)\cup\{A^1\}}(\varphi,\psi) = ... = \rho_{S(\Sigma_0)\cup\{A^m\}}(\varphi,\psi) = \rho_{S(\Sigma_1)}(\varphi,\psi)$.

ч.т.д.

Определение и свойства меры опровержимости

Подход к определению меры опровержимости основывается на естественном предположении: чем больше моделей на которых высказывание принимает значение не равное 1, тем высказывание легче опровержимо. Поскольку возможных значений не равных 1 несколько, то предлагаем учитывать их монотонными весами по всем этим значениям и для каждого такого значения истинности нормированными.

Перейдем к более формальному определению. Мера опровержимости $I_{S(\Sigma)}(\varphi)$ для формул из $\Phi(\Sigma) = \{\varphi \mid S(\varphi) \subset S(\Sigma)\}$ задается равенством

$$I_{S(\Sigma)}(\varphi) = \sum_{i=0}^{n-2} \alpha_i \frac{|\, Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}})\,|}{n^{|S(\Sigma)|}},$$

где α_i удовлетворяет условиям: $\begin{cases} 0 \leq \alpha_i \leq 1; \\ \alpha_i + \alpha_{n-1-i} = 1 \;\forall i = 0,...,\dfrac{n-1}{2}; \\ \alpha_k \geq \alpha_i \;\forall k \leq i; \end{cases}$

Лемма (свойства меры $I_{S(\Sigma)}$) Для любых формул $\varphi, \psi \in \Phi(\Sigma)$ справедливы

1) $0 \leq I_{S(\Sigma)}(\varphi) \leq 1;$
2) $I_{S(\Sigma)}(\varphi) + I_{S(\Sigma)}(\neg\varphi) = 1;$
3) $I_{S(\Sigma)}(\varphi \,\&\, \psi) \geq \max\{I_{S(\Sigma)}(\varphi), I_{S(\Sigma)}(\psi)\};$
4) $I_{S(\Sigma)}(\varphi \vee \psi) \leq \min\{I_{S(\Sigma)}(\varphi), I_{S(\Sigma)}(\psi)\};$
5) $I_{S(\Sigma)}(\varphi \vee \psi) + I_{S(\Sigma)}(\varphi \,\&\, \psi) = I_{S(\Sigma)}(\varphi) + I_{S(\Sigma)}(\psi);$

Доказательство:

1) Неравенство очевидно, так как. $Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}})$ попарно не пересекаются, а в объединении дают все множество $P(S(\Sigma))$. Свойство 1 доказано.

2) $I_{S(\Sigma)}(\varphi) + I_{S(\Sigma)}(\neg\varphi) =$

$$= \alpha_0 \frac{|\, Mod_{S(\Sigma)}(\varphi_0)\,|}{n^{|S(\Sigma)|}} + \alpha_{n-1} \frac{|\, Mod_{S(\Sigma)}(\varphi_1)\,|}{n^{|S(\Sigma)|}} + \sum_{i=1}^{n-2} (\alpha_i + \alpha_{n-1-i}) \frac{|\, Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}})\,|}{n^{|S(\Sigma)|}} = \frac{|\, P(S(\Sigma))\,|}{n^{|S(\Sigma)|}} = 1;$$

3) Распишем подробно, что такое $I_{S(\Sigma)}(\varphi \,\&\, \psi)$:

$$I_{S(\Sigma)}(\varphi \& \psi) = \sum_{i=0}^{n-2} \alpha_i \frac{|Mod_{S(\Sigma)}((\varphi \& \psi)_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{i=0}^{n-2} \alpha_i \left(\sum_{k=i}^{n-1} \left(\frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} \right) \right) - \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}};$$

Распишем подробно $I_{S(\Sigma)}(\varphi)$:

$$I_{S(\Sigma)}(\varphi) = \sum_{i=0}^{n-2} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} = \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{n-1} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{i=0}^{n-2} \alpha_i \sum_{k=i}^{n-1} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{i} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} - \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}};$$

$$I_{S(\Sigma)}(\varphi \& \psi) - I_{S(\Sigma)}(\varphi) = \sum_{i=0}^{n-2} \sum_{k=i}^{n-1} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} - \sum_{i=0}^{n-2} \sum_{k=i}^{n-1} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{i=0}^{n-2} \sum_{k=0}^{i} (\alpha_k - \alpha_i) \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \sum_{ki=0}^{n-2} \sum_{k=i}^{n-1} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} \geq 0;$$

Получили, что $I_{S(\Sigma)}(\varphi \& \psi) \geq I_{S(\Sigma)}(\varphi)$. По симметрии получим аналогичное неравенство для ψ:

$I_{S(\Sigma)}(\varphi \& \psi) \geq I_{S(\Sigma)}(\psi)$

=> $I_{S(\Sigma)}(\varphi \& \psi) \geq \max\{I_{S(\Sigma)}(\varphi), I_{S(\Sigma)}(\psi)\}$.

Свойство 3 доказано.

4) Распишем подробнее $I_{S(\Sigma)}(\varphi \vee \psi)$:

$$I_{S(\Sigma)}(\varphi \vee \psi) = \sum_{i=0}^{n-2} \alpha_i \frac{|Mod_{S(\Sigma)}((\varphi \vee \psi)_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{i=0}^{n-2} \alpha_i \left(\sum_{k=0}^{i} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \sum_{k=0}^{i} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} \right) -$$

$$- \sum_{i=0}^{n-2} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}};$$

Распишем подробнее $I_{S(\Sigma)}(\varphi)$:

$$I_{S(\Sigma)}(\varphi) = \sum_{i=0}^{n-2} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} = \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{n-1} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{i=0}^{n-2} \alpha_i \sum_{k=i}^{n-1} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{i} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} - \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \& \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}};$$

Вычислим разность двух полученных выше равенств:

$$I_{S(\Sigma)}(\varphi) - I_{S(\Sigma)}(\varphi \vee \psi) = \sum_{i=0}^{n-2} \alpha_i \sum_{k=i}^{n-1} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \,\&\, \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} + \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{i} \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} =$$

$$= \sum_{k=0}^{n-1}\sum_{k=0}^{k} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{i}{n-1}} \,\&\, \psi_{\frac{k}{n-1}})|}{n^{|S(\Sigma)|}} - \sum_{i=0}^{n-2} \alpha_i \sum_{k=0}^{i} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}} \geq \sum_{i=0}^{n-2}\sum_{k=0}^{i} \alpha_i \frac{|Mod_{S(\Sigma)}(\varphi_{\frac{k}{n-1}} \,\&\, \psi_{\frac{i}{n-1}})|}{n^{|S(\Sigma)|}};$$

Получили, что $I_{S(\Sigma)}(\varphi \vee \psi) \leq I_{S(\Sigma)}(\varphi)$. По симметрии получим аналогичное неравенство для ψ: $I_{S(\Sigma)}(\varphi \vee \psi) \leq I_{S(\Sigma)}(\psi)$

=> $I_{S(\Sigma)}(\varphi \vee \psi) \leq \min\{I_{S(\Sigma)}(\varphi), I_{S(\Sigma)}(\psi)\}$.

Свойство 4 доказано.

5) Из формул, полученных при доказательстве пунктов 3)- 4) непосредственно следует формула:

$I_{S(\Sigma)}(\varphi \vee \psi) + I_{S(\Sigma)}(\varphi \,\&\, \psi) = I_{S(\Sigma)}(\varphi) + I_{S(\Sigma)}(\psi)$.

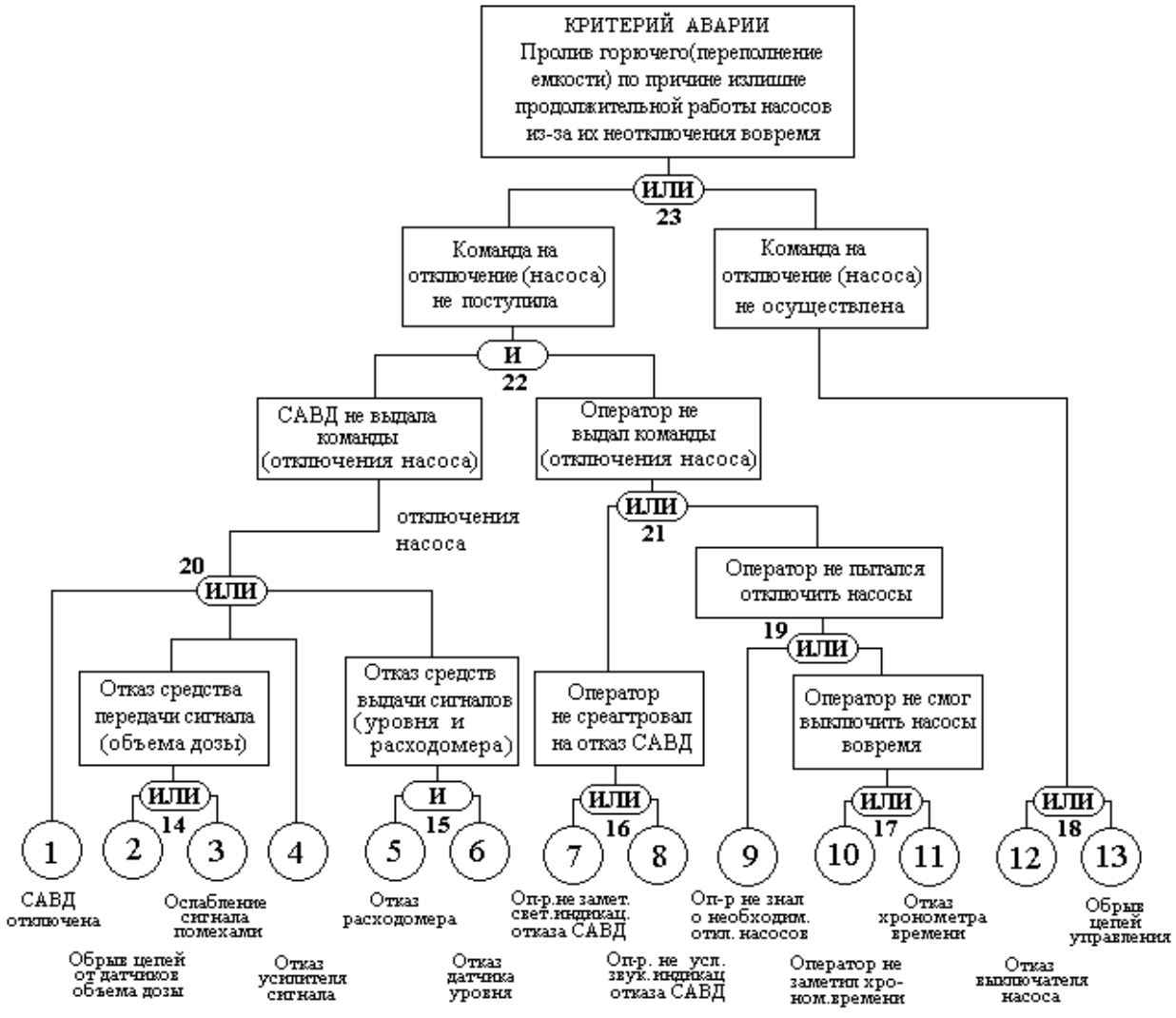


Рис.1. Дерево “отказа“ заправочной операции

Заметим, что связь меры опровержимости с введенным выше расстоянием не такая простая как в случаях логик при n=2 или 3. Для введения расстояния было перебрано много вариантов, для них желаемые свойства, с точки зрения экспертов, не выполнялись. Так что надеяться на другие случаи не приходится и можно констатировать, что в общем случае нет тесной связи между расстоянием и мерой опровержимости.

Рассмотрено дерево событий (рис.1), используемого для анализа причин возникновения аварийных ситуаций при автоматизированной заправке емкости. Структура дерева событий включает одно головное событие (авария, инцидент), которое соединяется с набором соответствующих нижестоящих событий (ошибок, отказов, неблагоприятных внешних воздействий), образующих причинные цепи (сценарии аварий).

Проанализированы из дерева различные высказывания об отказах заправочной станции и найдены расстояния между различными формулами и их меры опровержимости при различных n. Результаты показали адекватность предлагаемого подхода и качественную похожесть результатов на случаи n=2 и n=3. Для больших n ситуация такова: нами рассмотрено несколько задач по вычислению расстояния между формулами для различных n. Из полученных таблиц следует, что расстояния различны для различных n. И с ростом n расстояния все меньше и меньше отличаются друг от друга. Аналогичная ситуация со значениями меры опровержимости.

Заключение

Предложенные расстояния и мера опровержимости могут использоваться в изучении баз знаний, их пополнений, кластеризации, выявлению противоречивости, а так же в вопросах распознавания образов и адаптивных методах построения логических решающих функций по вероятностным высказываниям.

Литература

[1] Лбов Г.С., Старцева Н.Г. Логические решающие функции и вопросы статистической устойчивости решений. – Новосибирск: Изд-во Ин-та математики, 1999.

[2] Кейслер Г., Чэн Ч.Ч. Теория моделей. М.: Мир, 1977.

[3] Карпенко А.С. Логики Лукасевича и простые числа. –Москва: «Наука» 2000.

[4] G.Lbov, M.Gerasimov. Constructing of a Consensus of Several Experts Statements. In: Proc. of XII Int. Conf. ”Knowledge-Dialogue-Solution”, 2006, pp. 193-195.

[5] G.S. Lbov, M.Gerasimov. Interval Prediction Based on Experts’ Statements. In: Proc. of XIII Int. Conf. ”Knowledge-Dialogue-Solution”, 2007, Vol. 2, pp. 474-478.

[6] G.S.Lbov, M.K.Gerasimov. Determining of Distance Between Logical Statements in Forecasting Problems. In: Artificial Intelligence, 2’2004 [in Russian]. Institute of Artificial Intelligence, Ukraine.

[7] A.Vikent’ev. Measure of Refutation and Metrics on Statements of Experts (Logical Formulas) in the Models for Some Theory. In: Int. Journal ”Information Theories & Applications”, 2007, Vol. 14, No.1, pp. 92-95.

Author's Information

Vikent’ev Alexandr A. - *Institute of Mathematics, SB RAS, Akadem. Koptyug St., bl.4, Novosibirsk, Russia; e-mail: vikent@math.nsc.ru*

АДАПТИВНЫЕ ПОДХОДЫ К КОРРЕКЦИИ СТАТИЧЕСКИХ И КИНЕМАТИЧЕСКИХ ПОПРАВОК В ЗАДАЧЕ ОБРАБОТКИ СЕЙСМИЧЕСКИХ ДАННЫХ[1]

Татяна Ступина

***Аннотация**: В работе рассматриваются проблемы, возникающие в цифровой обработке сейсмических данных. Основной акцент делается на недостатки в изученности несоответствия математической и физической модели объекта. Не смотря на то, что любой граф обработки сейсмических данных содержит процедуру коррекции статических и кинематических поправок., учитывающих неоднородности верхней части сред, задача остаётся актуальной поскольку качество конечных результатов, представленных сейсмическими и глубинными разрезами и в настоящее время оставляет желать лучшего. Для регрессионной модели коррекции статики без искажения кинематики предлагается три подхода к реализации численного метода, максимально учитывающего априорную информацию, представленную количеством опорных точек на сейсмическом профиле.*

***Ключевые слова**: модель остаточных времен отражений, кинематические и статические поправки, линейная регрессия, мера априорной информации.*

***ACM Classification Keywords**: G1.10 Numerical Analysis - Applications, G3 Probability and statistics - Correlation and regression analysis.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Результаты обработки сейсмических данных зависят от того, насколько экспериментальные (полевые) данные соответствуют принятой теоретической модели изучаемой среды. Такие искажения, как правило, относятся к искажениям времён прихода волн за счет неоднородностей в верхней части разреза (статические поправки) и различий во временах прихода полезных отраженных волн, вызванных неодинаковым удалением пункта приёма и источника (кинематические поправки). К настоящему времени в современных обрабатывающих пакетах используются достаточно эффективные процедуры коррекции статических и кинематических поправок [В.С. Козырев и др] но, как правило, они основываются на предположении о вертикальности распространения лучей от уровня приведения до дневной поверхности. Такие предположения не всегда приводят к ожидаемому или удовлетворительному результату [А.П. Сысоев], поскольку не учитывается полная информация о неоднородности верхней части разреза, которую можно охарактеризовать:

1. зоной малых скоростей,
2. рельефом дневной поверхности,
3. погруженными неоднородностями.

Учет всех этих параметров в общей модели коррекции остаточных времен может дать более удовлетворительный результат.

Цель исследований проблемы учета поверхностных неоднородностей для задачи кинематической интерпретации точно определена в названии работы [В.М. Глоговский и др.] – получение неискаженных кинематических параметров временного поля, пересчитанного на линию приведения с учетом априорной информации о структуре рельефа дневной поверхности. Не смотря на большое количество работ по данной тематике, не существует достаточно чёткой постановки о количественной или качественной мере априорной информации необходимой для получения удовлетворительного результата. Это обусловлено

[1] Работа выполнена при поддержке РФФИ 07-01-00331-а

«сложностью» задачи - большим набором предопределяемых факторов: выбор модели покрывающей толщи и сейсмической границы, обобщение значений скоростей, система наблюдений и др.

В настоящей работе рассматривается линейная модель остаточных времен отражений [Л. Хаттон и др], проводится её анализ для задачи с фиксированной системой наблюдений, задаётся мера априорной информации, предлагается несколько адаптивных подходов к уточнению решения, т.е. учитывающих априорную информацию, основанную на введении комбинированных функционалов качества.

Модель остаточных времен отражений

Сначала будем рассматривать модель остаточных времен отражений в следующем представлении:

$$\tau_{ij} = S_i + R_j + G_k + M_k x_{ij}^2 + N_{ij} \quad (1)$$

где т – двойное время отражения после введения первичных априорных статических и кинематических поправок, *i*, *j*, *k* – индексы положения источника, приёмника и общей срединной точки (ОСТ) соответственно, *G* – структурная компонента (двойное время пробега волны от уровня приведения до отражающей границы в средней точке между источником и приёмником), *S* – поверхностно согласованная статическая поправка за источник (пункт возбуждения), *R* – то же за приёмник, *M* - коэффициент остаточной кинематической поправки, *x* – удаление приёмника от источника, *N* – компонента шума.

Модель (1) получается из уравнения относительно времен T_{ij} отраженной волны:

$$T(x,l) = a(x-l) + b(x+l) + \sqrt{t_0^2(x) + 4l^2 / v^2(x)}\,, \quad (2)$$

описывающей годограф ОСТ для однослойной модели среды, пересчитанный на линию приведения в системе координат (*x,l*): *x* - координата ОСТ, *l* - полурасстояние приемник-источник. Благодаря возможности получения и учета априорной информации, осуществляемой путем ввода априорной модели годографа $T_a(x,l)$ с параметрами $a_a(x-l)$, $b_a(x+l)$, t_{0a}, v_a, и сделав процедуру линеаризации модели (2) (разложение в ряд Тейлора функции $\sqrt{(\bullet)}$ в окрестности *l*=0), получим:

$$\tau(x,l) = T(x,l) - a_a(x-l) - b_a(x+l) - \sqrt{t_{0a}^2(x) + 4l^2 / v_a^2(x)} \approx$$

$$\approx s(x-l) + r(x+l) + g(x) + m(x)l^2 \quad (3)$$

модель, определяющую остаточные временные сдвиги $\tau(x,l) = \tau_{ij}$ в виде (1). Окончательным решением являются времена отражений

$$T(x,l) = T_a(x,l) + \tau(x,l). \quad (4)$$

Восстановление факторов, определяющих влияние параметров ВЧР на времена прихода волны отражения, по наблюдённым данным эквивалентно решению системы уравнений (1) минимизацией функционала $F_1 = \left\| \tau - \tilde{\tau} \right\|_{L_2}$. Отметим, что оценки факторов - мнк-оценки, которые обладают свойством оптимальности в случае независимости оцениваемых параметров и некоррелируемости шума, что не всегда выполнимо на практике. Так же при решении задач на практике возникают дополнительные трудности в реализации вычислительного алгоритма. Переписав уравнение (1) в операторной форме:

$$A\,p = \tau + \delta\tau, \quad (5)$$

можно увидеть, что матрица *A* сильно разреженная, состоящая из 0,1 и x_{ij}^2, $p = (S_i, R_j, G_k, M_k)^{\mathrm{T}}$ – вектор оцениваемых факторов, τ – вектор наблюдений, $\delta\tau$ – вектор помех. В связи с большой размерностью входных параметров и наличием между ними определённых связей система уравнений является плохо обусловленной, а решения - неустойчивыми [А.П. Сысоев]. При решении таких систем прибегают к различным методам регуляризации и итерационным методам решения систем уравнений, а также к привлечению дополнительной (априорной) информации, к упрощению моделей до меньшего числа параметров или сужая класс решающих функций. Однако, различные методы регуляризации не всегда являются эффективны в получении удовлетворительного конечного результата (временной или глубинный разрез среды), поскольку по своей идее направлены скорее на улучшение только алгоритмической части решения задачи.

Таким образом, не смотря на математическую простоту модели (1), возникают сопутствующие проблемы в её практической реализации и получении корректного результата. Видимо ещё и поэтому имеет смыл уделять должное внимание применению априорной информации об исследуемом объекте на всех этапах

решения задачи, которая влияет как на адекватность физической модели, так и на реализацию вычислительного алгоритма.

Экономическая актуальность решения обратных задач сейсмики и по настоящее время – в грамотном использовании априорной информации в связи с большими затратами на её получение.

Определим одну из количественных мер априорной информации через число μ опорных точек, необходимых для интерполирования формы $f_a(x)$ рельефа дневной поверхности. Будем пока считать, что все остальные параметры модели известны или имеются достаточно точные оценки. Сформулируем идейные подходы к решению обратной кинематической задачи с учётом априорной информации о рельефе дневной поверхности.

О подходах к решению обратной кинематической задачи с учётом априорной информации

При решении систем уравнений, подобных модели (1) с сильно разряженными матрицами на практике, как правило, применяются итерационные методы. Здесь возникает дополнительная задача – выбор устойчивого алгоритма решения в общем случае некорректной задачи. Неплохо зарекомендовал себя итерационный метод Гаусса-Зейделя [[Л. Хаттон и др.]. В целях исследования применим этот метод с учетом выполнения условий на решение, которое будет состоять в том, чтобы в определённых (опорных) точках, образующих множество D_a, функция рельефа f сохраняла бы значения близкие к априорно заданным (как правило, полученным в результате проведения работ по микросейсмокаратажу), т. е.

$$F_2 = \left\| f(x) - f_a(x) \right\|_{L_2} \to \min_{x \in D_a}.$$

По сути, можно сказать, что задача коррекции статических поправок без искажения кинематических факторов отражённых волн сводится к задаче адекватного восстановления многоэкстремальной функции рельефа по априорно заданным опорным точкам.

Покажем, что минимизация только функционала F_1 не всегда может привести к желаемому результату. Обозначим через g^*, f^* (параметры x и l опустим) истинные, по существу неизвестные функции годографа и рельефа, через g_a, f_a – априорные функции годографа и рельефа, через $\widetilde{g}, \widetilde{f}$ - кинематические и статические поправки. Тогда подставляя равенство (4) в уравнение функционала F_1 получим оценку

$$\left\| \tau - \widetilde{\tau} \right\| \le \left\| g^* - g_a \right\| + \left\| f^* - f_a \right\| + \left\| \widetilde{g} \right\| + \left\| \widetilde{f} \right\| = \varepsilon_g + \varepsilon_f + \left\| \widetilde{g} \right\| + \left\| \widetilde{f} \right\| = \varepsilon,$$

из которой видно, что одному значению критерия могут соответствовать различные значения входящих в сумму слагаемых. Даже при условии достаточно точно заданной априорной модели среды (два первых слагаемых, которые так же можно назвать неустранимыми погрешностями модели) кинематические и статические поправки могут быть взаимно заменяемыми.

Для того, чтобы скорректировать возникающую неоднозначность предлагается несколько идейных подходов при решении задачи:

1. Последовательная поправка априорного решения на каждом шаге итерационного метода решения системы уравнений (5).
2. Введение нового функционала качества, как линейной комбинации F_1 и F_2, т. е. $F = \sum \lambda_i F_i \to \min$.
3. Построение усредненного с весовым функциям решения на множестве «пробных» решений. Введение весовой функции на вектора решений с максимальным весом в априорно заданных точках рельефа.

Для оценки обобщающей способности алгоритма предлагается дополнительно исследовать норму невязки по контрольным точкам (исключенным из обучающей матрицы A). Хотя такой подход, как правило, применяется в задачах распознавания образов, автором не обнаружено его использование (или опровержение) в работах по обработке сейсмических данных.

В рассматриваемой задаче сильная разреженность операторной матрицы позволяет выписать итерационные формулы получения мнк-оценок для модели (1) не прибегая к непосредственному транспонированию:

$$G_k = \frac{1}{n_k}\sum_{(k)}^{n_k}(\tau_{ij} - S_i - R_j - M_k x_{ij}^2),$$

$$S_i = \frac{1}{n_i}\sum_{(i)}^{n_i}(\tau_{ij} - G_k - R_j - M_k x_{ij}^2),$$

$$R_j = \frac{1}{n_j}\sum_{(j)}^{n_j}(\tau_{ij} - S_i - G_k - M_k x_{ij}^2),$$

$$M_k = \frac{1}{\sum_{(k)}^{n_k} x_{ij}^2}\sum_{(k)}^{n_k}(\tau_{ij} - S_i - R_j - G_k), \qquad (6)$$

где n_k, n_i, n_j означают кратность, с которой *k*-ая ОСТ, *i*-ый пункт возбуждения и *j*-ый пункт приёма присутствуют в наблюдениях, суммирование ведётся по трассам, в которых они участвуют. Из представленных зависимостей (6) видно, что начальное приближение возможно по любому из факторов S_i, R_j, G_k, M_k .

Стратегия нахождения искомого решения состоит в реализации двух основных процедур:

1. Проверки условия сходимости итерационного процесса, например, по относительной невязке на *n* – ой итерации, либо по количеству допустимых итераций *n* > *maxiter*.
2. Проверки априорных условий, устанавливаемых предложенными подходами.

Пусть на *n*-ой итерации, $n = 1, 2, 3\ldots$ получено решение $p^{(n)} = (f^{(n)}, g^{(n)})^T$ системы $\tau^{(n)} = Ap^{(n)} = (A_1 f^{(n)}, A_2 g^{(n)})^T$, со значением невязки равным $\varepsilon^{(n)} = \left\|\tau - \tau^{(n)}\right\|$. Решение $p^{(n)}$ корректируется до $\widetilde{p}^{(n)} = p^{(n)} + p_1$ следующим образом.

1. Для первого подхода:

$\left\|\tau - \widetilde{\tau}^{(n)}\right\| \le \varepsilon^{(n)} + \varepsilon_1$ и $\left\|\widetilde{f}^{(n)}\right\| \le \left\|f^{(n)}\right\|$, где $\varepsilon^{(n)} + \varepsilon_1 \le \left\|\delta\tau\right\|$.

2. Для второго подхода:

$F = \lambda_1\left\|\tau - \widetilde{\tau}^{(n)}\right\| + \lambda_2\left\|\widetilde{f}^{(n)}\right\| \to \min\limits_{x\in D_a}$, где $\lambda_1 + \lambda_2 = 1$.

Условие $\lambda_2 > \lambda_1$ означает более сильные требования к выполнению априорных условий.

3. Для третьего подхода:

Обозначим через $D_p = \left\{p^{(n)} : \delta_{\min} \le \left\|\tau - Ap^{(n)}\right\| \le \delta_{\max}, n_{\min} \le n \le n_{\max}\right\}$ множество «пробных» решений, т.е. множество решений, которые в пределах нескольких итераций удовлетворяют допустимому качеству по норме невязки, тогда окончательное решение формируется как $\widetilde{p} = \sum_n \lambda_n p^{(n)}$, где $\sum_n \lambda_n = 1$ и весовой ряд $\{\lambda_n\}$ ранжирован с максимальным значением для минимальной нормы $\left\|f^{(n)}\right\|$.

Экспериментальное исследование устойчивости решения

При численном исследовании задачи на устойчивость возникают две составляющие, влияющие в целом на результат:

1. устойчивость численного алгоритма, когда желательно заранее оценить обусловленность матрицы *A*,

2. устойчивость самой модели (1), когда оценивается мера отклонения решения в зависимости от уровня шума $\left\|\delta\tau\right\|$.

На практике часто бывает трудно оценить число обусловленности матрицы *A*. Поэтому предлагается использовать один из приёмов, который хотя и не даёт строгий ответ об устойчивости применяемого

метода и полученного решения, но все же позволяет сделать некоторые предположения о характере решения. Такой подход несколько напоминает метод регуляризации решения и заключается в уточнении полученного решения и оценки его погрешности следующим образом.

Пусть получено приближенное решение системы (1) $\overline{p}^{(0)}$. Вычислим невязку уравнения $\overline{\xi} = \overline{\tau} - A\overline{p}^{(0)}$. Если норма невязки $\overline{\xi}$ велика, то можно искать вектор-поправку $\widetilde{p}$ такой, что точное решение системы $\overline{p} = \overline{p}^{(0)} + \widetilde{p}$. Следовательно $A \cdot (\overline{p}^{(0)} + \widetilde{p}) = \overline{p}$, отсюда $A\widetilde{p} = \overline{\xi}$. После этого уточняется решение системы $\overline{p}^{(1)} = \overline{p}^{(0)} + \widetilde{p}$. Если относительная погрешность $\frac{\|\widetilde{p}\|}{\|\overline{p}^{(1)}\|}$ велика, то можно повторить процесс уточнения. Если процесс уточнения, повторенный два три раза, не приводит к повышению точности, то это говорит скорей всего о том, что данная система плохо обусловлена и ее решение не может быть найдено с требуемой точностью.

Приведённый алгоритм экспериментального исследования устойчивости решения применим ко всем трём подходам, сформулированным выше, причём на этапе последовательного формирования решения.

Заключение

В настоящей работе представлена линейная модель остаточных времен отражений, учитывающая статические и кинематические поправки за рельеф с учётом априорной информации о самой модели. Было отмечено, что такого рода задачи остаются актуальными в связи с большой обусловленностью их решений. Предложено три адаптивных подхода к получению решения, т.е. максимально учитывающих априорную информацию, представленную количеством опорных точек на сейсмическом профиле. Предложенные подходы основаны на введении комбинированных функционалов качества модели и корректировки решения на этапе итерационного формирования решения. Для однородной однослойной изотропной среды заданной мощности, фиксированной системы наблюдений определённой длины, планируется численное моделирование по представленной в работе методике.

В заключении отметим, что кроме применения математического аппарата, используемого для четкого обоснования методов, любая практическая задача всегда требует индивидуального подхода к её решению, не смотря на общепринятые подходы.

Литература

[В.С. Козырев и др.] В.С. Козырев, А.П. Жуков, И.П. Коротков, А.А. Жуков, М.Б. Шнеерсон. Учёт неоднородностей верхней части разреза в сейсморазведке. Современные технологии, М.: ООО «Недра-Бизнесцентр», 2003, 227 с.

[А.П. Сысоев] А.П. Сысоев. Анализ устойчивости оценивания статических и кинематических параметров в МОВ. Математические проблемы интерпретации данных сейсморазведки, Новосибирск, Наука, 1988.

[Л. Хаттон и др.] Л. Хаттон, М. Уэрдингтон, Дж. Мейкин. Обработка сейсмических данных. Теория и практика. Изд-во: М.: Мир, 1989, 216 с.

[В.М. Глоговский] В.М. Глоговский, А.Р. Хачатрян. Коррекция статических поправок без искажения кинематических параметров отражённых волн. Геология и геофизика. 1984, №10,

Информация об авторе

Tatyana A. Stupina *– Trofimuk Institute of Petroleum-Gas Geology and Geophisics of SBRAS, Koptyug Ave 3, Novosibirsk, 630090, Russia; e-mail: stupinata@ipgg.nsc.ru*

СИНТЕЗ УРАВНЕНИЙ УПРАВЛЕНИЯ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ РОБОТОВ

Юрий Кук, Елена Лаврикова

***Abstract**: The equations of management for intelligent robots are synthesized. The received equations are used by the robot for optimum transformation of an initial situation to the necessary target situation.*

***Keywords**: situation, equations of management, intelligent robots.*

***ACM Classification Keywords**: I.2.8 Problem Solving, Control Methods, and SearchH.1.1 Systems and Information.*

***Conference**: The paper is selected from XIVth International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008*

Введение

Система управления роботом имеет иерархическую структуру. На нижнем, первом, уровне осуществляется управление непосредственно движением исполнительных механизмов робота. На втором уровне формируются сигналы управления. На этом уровне управление распределяется по всем исполнительным механизмам робота для осуществления требуемой элементарной операции. На третьем уровне осуществляется расчленение более крупной операции на элементарные операции и составляется последовательность их выполнений в соответствии с некоторым набором правил. Сигнал о необходимости выполнения каждой из элементарных операций поступает из третьего во второй уровень системы управления. Четвертый уровень управления реализуется в интеллектуальных роботах. Этот уровень необходим в тех случаях, когда заранее неизвестно, какую операцию нужно выполнить. Робот, исходя из окружающей ситуации, которая для него неизвестна и которая может изменяться, должен сам принять решение какую операцию необходимо реализовать. Следовательно, четвертый уровень – это уровень выработки и принятий решения о необходимости выполнения той или иной операции в заранее неопределенных условиях. Принятое решение передается в третий уровень системы управления для реализации. В докладе рассматриваются роботы с четвертым уровнем управления. На этом уровне предлагается следующая процедура управления. Роботы строят математическую модель ситуации, в которой они находятся и на основе этой модели синтезируют уравнения управления для получения управляющей информации. Эта информация затем используется для оптимального преобразования ими исходной ситуации в нужную целевую ситуацию.

Управление роботами включает следующие четыре стадии: обучение, запоминание программы, воспроизведение программы и отработку программы. Обучение робота осуществляет человек различными путями. Полученная в процессе обучения информация запоминается роботом в виде программы в течение заданного времени. Воспроизведение программы осуществляется путем считывания информации из устройства памяти и передачи управляющих сигналов к исполнительным механизмам робота. В интеллектуальных роботах возможно изменение записанной в памяти робота информации в соответствии с конкретной ситуацией. Отработка программы заключается в выполнении роботом рабочих операций по сигналам, переданным его исполнительным механизмом при воспроизведении программы.

Каждый уровень системы управления робота имеет обратные связи, по которым передается информация о состоянии и действии нижних уровней.

Построение модели ситуации

Автоматическое управление интеллектуальным роботом будем осуществлять на основе построенной им математической модели той ситуации, в которой он находится. Модель строится на основе данных, полученных роботом за определенный промежуток времени. Построенная модель позволяет правильно спрогнозировать последующие его действия для преобразования исходной ситуации в нужную целевую

ситуацию. Будем придерживаться определения ситуации, которое предложил В.П. Гладун в [1]. Ситуация рассматривается им как фрагмент среды. Среда определяется как тройка $\langle V, K, A \rangle$, где V – множество объектов, K – множество свойств, состояний и связей объектов из множества V. A – множество действий, которые можно выполнять с элементами множеств V и K. Под ситуацией понимается пара $\langle V_s, K_s \rangle$, где V_s и K_s – подмножества соответственно множеств V и K.

Будем считать, что ситуация состоит из конечного числа объектов, свойства, состояния и связи между которыми описываются множеством переменных. Выделим следующие виды переменных. Переменные, описывающие ситуацию, и значения которых могут изменяться при действиях робота с целью преобразовании им исходной ситуации в целевую ситуацию, будем называть ситуационными и обозначать – $(y_1,...,y_Q)$. Действия робота осуществляются под воздействием управляющих сигналов. Переменные, описывающие значения управляющих сигналов робота, называются управляющими – $(u_1,...,u_K)$. Пусть $\varepsilon_1..,\varepsilon_Q$, $\zeta_1,...,\zeta_N$ — случайные возмущения, не зависящие от времени и воздействующие соответственно на ситуационные и управляющие переменные. Без потери общности предположим, что случайные возмущения, воздействующие на ситуационные и управляющие переменные, имеют нулевые средние: $M\varepsilon_i = 0$, $i = 1,\ldots,Q$, $M\zeta_k = 0$, $k = 1,\ldots,K$. Пусть известна ковариационная матрица R случайных ошибок $\varepsilon_1..,\varepsilon_Q$, $\zeta_1,...,\zeta_N$:

$$R_i = \begin{pmatrix} \sigma^2(\varepsilon_1) & r(\varepsilon_1,\varepsilon_2) & \ldots & r(\varepsilon_1,\zeta_N) \\ r(\varepsilon_2,\varepsilon_1) & \sigma^2(\varepsilon_2) & \ldots & r(\varepsilon_2,\zeta_N) \\ \ldots & \ldots & \ldots & \ldots \\ r(\zeta_N,\varepsilon_1) & r(\zeta_N,\varepsilon_2) & \ldots & \sigma^2(\zeta_N) \end{pmatrix}. \quad (1)$$

На диагонали этой матрицы стоят дисперсии возмущений. Для построения модели ситуации будем предполагать, что значения всех рассматриваемых переменных наблюдались роботом в моменты времени $t = 1,\ldots,n$. Предположим, что в различные моменты времени возмущения между собой не коррелированны. С целью учета изменений, происходящих в ситуации при ее преобразовании роботом, модель будем строить в виде системы разностных уравнений, в левых частях которых стоят приращения ситуационных и управляющих переменных, а в правых – некоторые функции от ситуационных и управляющих переменных.

В основу метода построения модели ситуации положен подход, позволяющий избежать большого перебора моделей, и основанный на следующем принципе: в модель данного порядка должны входить переменные, имеющие только значимые значения частного коэффициента корреляции соответственно с приращениями ситуационной или управляющей переменной, что значительно упрощает модель при сохранении ее корректности. Множество таких переменных получается с помощью аппарата бассейнов и имеет аналогию с эффективным множеством регрессоров в регрессионном анализе [2]. Из исходного принципа вытекает следующая особенность предлагаемого метода: перебор моделей в пределах множества полиномиальных моделей одного и того же порядка не производится: оптимальная модель данного порядка находится с помощью построения бассейна, а перебор осуществляется среди оптимальных моделей разных порядков, что позволяет резко сократить общий перебор всех моделей. Принцип отбора переменных, попадающих в бассейн, имеет аналогию с принципом включения переменных в шаговом регрессионном методе [2]. Однако оба метода используют разные статистические критерии проверки значимости переменных, причем применение шагового метода требует построения промежуточных моделей с использованием метода наименьших квадратов (МНК), который принципиально не применим в нашем случае, поскольку аргументы модели подвержены случайным возмущениям. Если использовать МНК, то получатся модели, не соответствующие истинным зависимостям между переменными.

В результате случайных возмущений наблюдаются не истинные значения переменных, а значения, отличающиеся от них на некоторую случайную величину. Будем обозначать фактические значения

переменных, то есть такие, которые наблюдаются — с волной, а истинные значения переменных – со штрихом. Тогда

$$\widetilde{y}_1 = y'_1 + \varepsilon_1, ,..., \widetilde{y}_Q = y'_Q + \varepsilon_Q,\ \widetilde{u}_1 = u'_1 + \varsigma_1,..., \widetilde{u}_K = u'_K + \varsigma_K \quad (2)$$

Задача состоит в построении по экспериментальным данным о наблюдаемых значениях переменных математической модели ситуации, которая бы учитывала искажающее воздействие случайных возмущений на эти переменные. Пусть связь между приращениями ситуационных и управляющих переменных и значениями всех переменных, описывающих ситуацию, функциональна и описывается системой уравнений:

$$\begin{aligned} &\frac{\Delta y'_1}{\Delta t} = f_1(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,,\ \ldots\ldots\,,\ \frac{\Delta y'_Q}{\Delta t} = f_Q(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K) \\ &\frac{\Delta u'_1}{\Delta t} = f_{Q+1}(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,,\ldots\ldots\,,\ \frac{\Delta u'_K}{\Delta t} = f_{Q+K}(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K) \end{aligned} \quad (3)$$

Чаще всего функции f_i, $i = 1,\ldots,Q$, в (3) неизвестны. Поэтому ищется их аппроксимация с помощью каких-нибудь простых математических функций. Под моделью ситуации будем понимать систему соотношений

$$\begin{aligned} &\frac{\Delta y'_1}{\Delta t} = F_1(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,,\ \ldots\ldots\,,\ \frac{\Delta y'_Q}{\Delta t} = F_Q(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,, \\ &\frac{\Delta u'_1}{\Delta t} = F_{Q+1}(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,,\ \ldots\ldots\,,\ \frac{\Delta u'_K}{\Delta t} = F_{Q+K}(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)\,, \end{aligned} \quad (4)$$

где $F_i(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)$ – аппроксимация неизвестной функции $f_i(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)$, $i = 1,\ldots,Q,Q+1,\ldots,Q+K$.

Ограничимся рассмотрением полиномиальных моделей. В полиномиальной модели k-го порядка функция $F_i(y'_1,\ldots,y'_Q,u'_1,\ldots u'_K)$ представляет собой отрезок ряда Тейлора до производных $k+1$-го порядка. Любую полиномиальную модель можно представить в виде линейной модели:

$$\frac{\Delta y'_i}{\Delta t} = b_{i,0} + \sum_{j=1}^{L_k} b_{i,j} z_j\,,\ i = 1,\ldots,Q\,,\ \frac{\Delta u'_l}{\Delta t} = b_{l,0} + \sum_{j=1}^{L_k} b_{l,j} z_j\,,\ l = 1,\ldots,K \quad (5)$$

где L_k — равно числу всех членов аппроксимирующего отрезка ряда Тейлора, содержащих переменные, z_j обозначает переменную, либо произведение переменных, для j-го такого члена, а $b_{i,j}$ — соответствующий коэффициент. Таким образом, изучение полиномиальных моделей сводится к изучению линейных моделей (5). В отличие от МГУА [3], в котором переменные модели находятся многорядной селекцией, или при малом их числе — полным перебором, множество переменных, входящих в модель (5) находится с помощью бассейнов.

Бассейн для переменной $\Delta y'_i$ представляет собой набор таких переменных из множества $y'_1,\ldots,y'_Q,u'_1,\ldots u'_K$, которые имеют значимую связь с переменной $\Delta y'_i$. Аналогичным образом определяется бассейн для переменной $\Delta u'_i$. Для измерения связи между переменными будем использовать частные коэффициенты корреляции, которые измеряют статистическую связь между переменными, «очищенную» от опосредованного влияния других переменных. Под значимостью этой связи будем понимать значимость (с вероятностью ошибки α) значения частного коэффициента корреляции между ними [2]. Наилучшим множеством переменных для переменной $\Delta y'_i$ понимается набор таких переменных из множества $y'_1,\ldots,y'_Q,u'_1,\ldots u'_K$, которые обладают следующими свойствами: 1) полнотой: все переменные, имеющие с заданной вероятностью ошибки значимую связь с переменной $\Delta y'_i$, входят в это множество; 2) отсутствием избыточности: переменные, имеющие с заданной

вероятностью ошибки незначимую связь с переменной $\Delta y_i'$, не входят в это множество. Аналогичным образом определяется наилучшее множество для переменной $\Delta u_i'$. Наилучшие множества для переменных $\Delta y_i'$ и $\Delta u_i'$ находятся путем построения бассейнов. Бассейном k-го порядка для переменной $\Delta y_i'$ назовем подмножество B ситуационных и управляющих переменных, а также всевозможных их произведений (с числом сомножителей не более k), включающее все переменные, которые имеют значимый (относительно множества B) частный выборочный коэффициент корреляции с $\Delta y_i'$. Аналогичным образом определяется бассейн k-го порядка для переменной $\Delta u_i'$.

Всегда существует бесконечное число функций, построенных по экспериментальным данным, которые можно взять в качестве модели для истинной зависимости между переменными. Поэтому возникает вопрос: как наилучшим образом использовать экспериментальные данные для получения в каком-то роде оптимальной модели и что под этим следует понимать. Оптимальной моделью для $\Delta y_i'$ назовем модель, построенную из переменных наилучшего множества, для которой сумма квадратов отклонений функции $F_i(y_1',\ldots,y_Q',u_1',\ldots u_K')$ от истинной функции $f_i(y_1',\ldots,y_Q',u_1',\ldots u_K')$ минимальна по сравнению с другими моделями:

$$\min_{F_i}\sum_{t=1}^{n}[f_i(y_1',\ldots,y_Q',u_1',\ldots u_K')-F_i(y_1',\ldots,y_Q',u_1',\ldots u_K')]^2. \qquad (6)$$

Будем считать, что коэффициенты при переменных в выражении $F_i(y_1',\ldots,y_Q',u_1',\ldots u_K')$ формулы (6), не вошедшие в бассейн B, равны нулю. Аналогичным образом определяется оптимальная модель для переменной $\Delta u_i'$. Рассмотрим методику построения оптимальной линейной модели для переменной $\Delta y_i'$. Пусть в бассейн первого порядка для переменной $\Delta y_i'$ вошли следующие переменные: $y_{l1}',\ldots,y_{l\nu}',u_{k1}',\ldots,u_{kw}'$, которые составляют наилучшее множество переменных. Представим оптимальную линейную модель для переменной $\Delta y_i'$ в виде:

$$\frac{\Delta y_i'}{\Delta t}=b_{i,0}+b_{i,1}y_{l1}'+\ldots+b_{i,\nu}y_{l\nu}'+b_{i,\nu+1}u_{k,1}'+\ldots+b_{i,\nu+w}u_{k,w}', \qquad (7)$$

Найдем выражения для коэффициентов этой модели, чтобы выполнялось условие оптимальности модели. Для определения коэффициентов $b_{i,0},b_{i,1},\ldots,b_{i,\nu+w}$ в регрессионных методах используется метод наименьших квадратов для наблюдаемых переменных, при котором минимизируется следующая сумма квадратов:

$$\sum_{t=1}^{n}\{\frac{\Delta\tilde{y}_i}{\Delta t}-(b_0+b_{i,1}\tilde{y}_{l1}+\ldots+b_{i,\nu}\tilde{y}_{l\nu}+b_{i,\nu+1}\tilde{u}_{k1}\ldots+b_{i,\nu+w}\tilde{u}_{kw})\}^2. \qquad (8)$$

В случае, когда входные переменные подвержены случайным ошибкам, применение этого метода приводит к ошибочным моделям по сравнению с истинными зависимостями между переменными.

Поэтому коэффициенты модели (7) определяются не путем решения системы нормальных уравнений, которая получается в результате минимизации вышеуказанной суммы квадратов (8), а путем решения системы уравнений, которая получается в результате минимизации суммы квадратов (6):

$$\sum_{t=1}^{n}[\frac{\Delta y_i'}{\Delta t}-(b_{i,0}+b_{i,1}y_{l1}'+\ldots+b_{i,\nu}y_{l\nu}'+b_{i,\nu+1}u_{k1}'\ldots+b_{i,\nu+w}u_{kw}')]^2. \qquad (9)$$

Представим выражение в квадратных скобках (9) через наблюдаемые переменные и ошибки по формулам (2). Получим

$$\sum_{t=1}^{n}\{\frac{\tilde{y}_i(t+\Delta t)-\tilde{y}_i(t)-\varepsilon(t+\Delta t)+\varepsilon(t)}{\Delta t}= \qquad (10)$$

$$-(b_{i,0}+b_{i,1}\widetilde{y}_{l1}+\ldots+b_{i,\nu}\widetilde{y}_{l\nu}+b_{i,\nu+1}\widetilde{u}_{k1}\ldots+b_{i,\nu+w}\widetilde{u}_{kw})-$$
$$-(b_{i,1}\varepsilon_{l1}+\ldots+b_{i,\nu}\varepsilon_{l\nu}+b_{i,\nu+1}\varsigma_{k1}\ldots+b_{i,\nu+w}\varsigma_{kw})\}^2$$

Так как коэффициенты $b_{i,0},b_{i,1},\ldots,b_{i,\nu+w}$ доставляют минимум (6), то они удовлетворяют системе уравнений, получаемой из выражения (10) путем его дифференцирования по каждому из коэффициентов $b_{i,0},b_{i,1},\ldots,b_{i,\nu+w}$ с последующим приравниванием производных к нулю. В полученной системе уравнений выборочные коэффициенты ковариаций ошибок заменим истинными коэффициентами ковариаций ошибок из матрицы (1), а выборочные средние заменим истинными средними, т.е. – нулями. В результате будем иметь следующую систему уравнений. Первое уравнение этой системы равно:

$$\frac{\Delta\overline{y}_i}{\Delta t}=b_{i,0}+b_{i,1}\overline{y}_{l1}+\ldots+b_{i,\nu}\overline{y}_{l\nu}+b_{i,\nu+1}\overline{u}_{k1}\ldots+b_{i,\nu+w}\overline{u}_{kw}, \tag{11}$$

где $\overline{y}_i=\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_i(t),\quad \overline{y}_{l1}=\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l1}(t),\ldots,\overline{y}_{l\nu}=\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l\nu}(t),\ \ldots,\ \overline{u}_{kw}=\frac{1}{n}\sum_{t=1}^{n}\widetilde{u}_{kw}(t).$

Второе уравнение системы имеет вид:

$$\begin{aligned}
&\sum_{t=1}^{n}\frac{\Delta\widetilde{y}_i}{\Delta t}\widetilde{y}_{l1}=b_{i,0}+b_{i,1}\{\frac{1}{n}\sum_{t=1}^{n}[\widetilde{y}_{l1}(t)]^2-r(\varepsilon_{l,1},\varepsilon_{l,1})\}+\ldots\\
&\ldots+b_{i,2}\{\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l1}(t)\widetilde{y}_{l2}(t)-r(\varepsilon_{l,1},\varepsilon_{l,2})\}+\ldots\\
&\ldots+b_{i,\nu}\{\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l1}(t)\widetilde{y}_{l\nu}(t)-r(\varepsilon_{l,1},\varepsilon_{l,\nu})\}+\ldots\\
&\ldots+b_{i,\nu+w}\{\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l1}(t)\widetilde{u}_{kw}(t)-r(\varepsilon_{l,1},\varsigma_{k,w})\}
\end{aligned} \tag{12}$$

И, наконец, последнее уравнение равно:

$$\begin{aligned}
&\sum_{t=1}^{n}\frac{\Delta\widetilde{y}_i}{\Delta t}\widetilde{u}_{k,w}=b_{i,0}+b_{i,1}\{\frac{1}{n}\sum_{t=1}^{n}[\widetilde{y}_{l,1}(t)\widetilde{u}_{k,w}]-r(\varepsilon_{l,1},\varsigma_{k,w})\}+\\
&+b_{i,2}\{\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l,1}(t)\widetilde{u}_{k,w}(t)-r(\varepsilon_{l,1},\varsigma_{k,w})\}+\ldots\\
&\ldots+b_{i,\nu}\{\frac{1}{n}\sum_{t=1}^{n}\widetilde{y}_{l,\nu}(t)\widetilde{u}_{k,w}-r(\varepsilon_{l,\nu},\varsigma_{k,w})\}+\ldots\\
&\ldots+b_{i,\nu+w}\{\frac{1}{n}\sum_{t=1}^{n}[\widetilde{u}_{k,w}(t)]^2-r(\varsigma_{k,w},\varsigma_{k,w})\}
\end{aligned}$$

Неизвестные коэффициенты $b_{i,0},b_{i,1},\ldots,b_{i,\nu+\mu+w}$ легко находится из (11)-(13) по правилу Крамера. Аналогичным образом строится оптимальная модель для переменной $\Delta u_i'$.

В силу (5) методика построения оптимальных полилинейных моделей для переменных $\Delta y_i'$ и $\Delta u_i'$ аналогична методике построения оптимальных линейных моделей, но уже с использованием бассейнов k-го порядка для переменных $\Delta y_i'$ и $\Delta u_i'$ с учетом не только ковариаций ошибок, но и их смешанных моментов более высокого порядка. Построив модель ситуации (7) на основе экспериментальных данных, полученных за первые n тактов времени $t=1,\ldots,n$, можно на основе этой модели спрогнозировать значения ситуационных и управляющих переменных в следующий $n+1$ такт времени. Обозначим

найденные таким образом прогнозные значения ситуационных и управляющих переменных соответственно через $y_{1,progn}(n+1),\ldots,y_{Q,progn}(n+1)$ и $u_{1,progn}(n+1),\ldots,u_{K,progn}(n+1)$. Модель ситуации (7) назовем прогнозирующей. Получив прогнозные значения ситуационных и управляющих переменных в $n+1$ такт времени, строится так называемая модель управления ситуацией. Ее построение аналогично предыдущей модели. Назначение этой модели найти оптимальные значения приращений управляющих переменных в $n+1$ такт времени. В отличие от прогнозирующей модели в этой модели в качестве дополнительных переменных используются ситуационные и управляющие переменные для момента времени $t=n+1$, а в качестве их значений при определении коэффициентов модели – прогнозные значения ситуационных и управляющих переменных в $n+1$ такт времени, полученных из модели ситуации. В модели управления ситуацией будем также учитывать предысторию ситуации, то есть значения ситуационных и управляющих воздействий за последние два или несколько тактов времени. Обозначим предпоследние два момента времени $t=n-1$, $t=n-2$. Пусть $\Delta t=1$. Модель управления ситуацией – это система из K уравнений, которые будем также называть уравнениями управления. l -ое уравнение этой системы имеет следующий вид

$$\begin{aligned}
&\Delta u'_l(n)=b_{l,0}(n)+b_{l,1}y'_1(n)+\ldots+b_{l,Q}y'_Q(n)+\\
&+b_{l,Q+1}u'_1(n)+\ldots+b_{l,Q+K}u'_K(n)+b_{l,Q+K+1}y'_1(n-1)+\ldots+b_{l,2Q+K}y'_Q(n-1)+\\
&+b_{l,2Q+K+1}u'_1(n-1)+\ldots+b_{l,2Q+2K}u'_K(n-1)+b_{l,2Q+2K+1}y'_1(n-2)+\ldots+b_{l,3Q+2K}y'_Q(n-2)+\\
&+b_{l,3Q+2K+1}u'_1(n-2)+\ldots+b_{l,3Q+3K}u'_K(n-2)+b_{l,3Q+3K+1}y'_1(n+1)+\ldots+b_{l,4Q+3K}y'_Q(n+1)+\\
&+b_{l,4Q+3K+1}u'_1(n+1)+\ldots+b_{l,4Q+4K}u'_K(n+1),\ldots,l=1,\ldots,K.
\end{aligned} \tag{14}$$

Для всех переменных, фигурирующих в левой части (14), находятся частные коэффициенты корреляции с переменной $\Delta u'_i(n)$. После чего коэффициенты при переменных в левой части (14), имеющих незначимые частные коэффициенты корреляции с переменной $\Delta u'_i(n)$, обнуляются. Далее находятся значения оставшихся коэффициентов системы (14) аналогично тому, как это было проделано для системы уравнений (7). Коэффициенты находятся из уравнений аналогичных (11), (12) и (13), в которых используются прогнозные значений ситуационных и управляющих переменных $y_{1,progn}(n+1),\ldots,y_{Q,progn}(n+1)$, $u_{1,progn}(n+1),\ldots,u_{K,progn}(n+1)$ в $n+1$ такт времени. Определив коэффициенты уравнения (14), следующим шагом является нахождение оптимальных управляющих воздействий $u'_1(n+1),\ldots,u'_K(n+1)$ в $n+1$-й такт времени. Задача облегчается тем, что уравнение (14) линейно относительно управляющих воздействий. Вследствие чего, управляющие воздействия за один такт времени имеют верхнюю и нижнюю границу по своей величине. Поэтому, используя полный перебор всевозможных максимальных и минимальных значений этих воздействий, можно выбрать требуемые управляющие воздействия $u'_1(n+1),\ldots,u'_K(n+1)$, которые обеспечивали максимальное приближение ситуационных переменных к требуемым их значениям. В момент времени $t=n+1$ робот реализует эти воздействия, при этом ситуационные переменные принимают некоторые новые значения, поступающие на датчики робота. Таким образом, к моменту времени $t=n+2$ имеются новые экспериментальные данные об ситуационных и управляющих переменных. На основе этих данных и, используя прежние данные за период времени $t=2,\ldots,n$, строятся новые модели: прогнозирующая модель и модель управления ситуацией, аналогично как это было проделано выше. Такая процедура повторяется со сдвигом на один такт до тех пор, пока робот не преобразует исходную ситуацию в требуемую ситуацию. Для построения более точной прогнозирующей модели можно использовать ситуационные и управляющие переменные за последние два такта времени, каждая из которых выступает в роли самостоятельной переменной:

$$\Delta y_i'(n) = b_{i,0}(n) + b_{i,1}y_1'(n) + \ldots + b_{i,Q}y_Q'(n) +$$
$$+ b_{i,Q+1}u_1'(n) + \ldots + b_{i,Q+K}u_K'(n) + b_{i,Q+K+1}y_1'(n-1) + \ldots + b_{i,2Q+K}y_Q'(n-1) +$$
$$+ b_{i,2Q+K+1}u_1'(n-1) + \ldots + b_{i,2Q+2K}u_K'(n-1) + b_{i,2Q+2K+1}y_1'(n-2) + \ldots + b_{i,3Q+2K}y_Q'(n-2) +$$
$$+ b_{i,3Q+2K+1}u_1'(n-2) + \ldots + b_{i,3Q+3K}u_K'(n-2), \quad i = 1, \ldots, Q.$$

$$\Delta u_l'(n) = b_{l,0}(n) + b_{l,1}y_1'(n) + \ldots + b_{l,Q}y_Q'(n) +$$
$$+ b_{l,Q+1}u_1'(n) + \ldots + b_{l,Q+K}u_K'(n) + b_{l,Q+K+1}y_1'(n-1) + \ldots + b_{l,2Q+K}y_Q'(n-1) +$$
$$+ b_{l,2Q+K+1}u_1'(n-1) + \ldots + b_{l,2Q+2K}u_K'(n-1) + b_{l,2Q+2K+1}y_1'(n-2) + \ldots + b_{l,3Q+2K}y_Q'(n-2) +$$
$$+ b_{l,3Q+2K+1}u_1'(n-2) + \ldots + b_{l,3Q+3K}u_K'(n-2), \quad l = 1, \ldots, K.$$

В отличие от уравнений управления в прогнозирующей модели не используются управляющие переменные для будущего момента времени.

Выводы

В докладе предложено на четвертом уровне системы управления интеллектуальными роботами строить прогнозирующую модель ситуации, в которой он находится, а также на ее основе синтезировать уравнения управления, или модель управления ситуацией. Модель ситуации строится роботом на основе данных, полученных им за некоторый промежуток времени наблюдения за ситуацией. Построенная модель ситуации позволяет правильно спрогнозировать изменение ситуационных переменных в последующий момент времени. Имея в своем распоряжении спрогнозированные значения ситуационных и управляющих переменных, робот синтезирует линейные уравнения управления. На основе этих уравнений путем полного перебора максимальных и минимальных возможных значений управляющих воздействий за один такт времени робот находит для очередного момента времени значения оптимальных управляющих сигналов для исполнительных механизмов. Оптимальность понимается в том смысле, что обеспечивается максимальное приближение ситуационных переменных к требуемым их значениям. В результате действий робота изменяются значения ситуационных переменных в последующий момент времени. Эти измененные значения ситуационных переменных используются для построения новой модели ситуации в очередной момент времени. Процедура повторяется со сдвигом на один такт, пока робот не преобразует исходную ситуацию в требуемую ситуацию.

Библиография

1. Гладун В.П. Планирование решений. Киев: Наук. Думка, 1987.168с.
2. Вучков И., Бояджиева Л., Солаков Е. Прикладной линейный регрессионный анализ. Пер. с болг. – М.: Финансы и статистика, 1987. — 239с.
3. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами. Киев: Техника, 1975. 312с.

Authors' Information

Yurij Kuk – *The Institute of Cybernetics of National Academy of Science of the Ukraine, the senior scientist, address: 40 Glushkov ave., Kiev, Ukraine, 03680; e-mail: 1913@i.com.ua*

Helen Lavrikova – *The Institute of Cybernetics of National Academy of Science of the Ukraine, address: 40 Glushkov ave., Kiev, Ukraine, 03680; e-mail: icdepval@ln.ua*