723

ISSN 0868-4081
0868-4200

TALLINNA TEHNIKAÜLIKOOLI

TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

ТРУДЫ ТАЛЛИННСКОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

DATA PROCESSING

COMPILER WRITING

PROBLEMS OF PROGRAMMING

TALLINN 1991

UDC 681.06
681.3.06
681.3.016
519.4
519.8+519.6

Department of Data Processing 53 29 35
Raja 15
Tallinn
EE0108 Estonia

Editor I. Amitan

TTÜ, 1992, 350, 146
Rbl. 11.60

723

ALUSTATUD 1937

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

ТРУДЫ ТАЛЛИНСКОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

C. Informatics. Control Engineering

DATA PROCESSING
COMPILER WRITING
PROBLEMS OF PROGRAMMING

Transactions of the Faculty of Economics LXXV

TALLINN 1991

Contents

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.06

L. Vyhandu, R. Kuusik

EXPERT SYSTEM AS A PART OF INFORMATION SYSTEM

To use computers effectively in man-machine interactive environments one has to create proper algorithms and construct a user-friendly environment in a computer system.

Our special interest lies in cases of scarce and incomplete information for control. Industrial processes serve as our example, where physical and chemical processes are so complex that mathematical modeling is out of the question. At the same time the general principles of their control are well known and already in practical use.

For these problems we need formal tools to collect, represent and use knowledge of human experts.

In practice an expert concentrates on parameters of the controlled object which are influencing the control goal. From the observed values he makes necessary fuzzy conclusions and changes the values of control parameters.

We are searching a type of expert's inference model that would define the sets of conditions which should be satisfied to start a set of actions. Therefore, we have to use a model which would check conditions that hold and knows what real actions comply with them. That brings us to a natural way of describing the experts's knowledge. One has to construct a set of conditional productions, each of them having the form

IF (set of conditions) THEN (set of actions).

It is easy to understand the semantics of that production representing the dependence between the set of conditions to be fulfilled so that the specific actions can be taken.

To create all conditional productions needed for expert actions, we use the theory of monotonic systems [2].

TABLE 1

Parameters

conditional					decisional	
1	2	3	4	5	6	7
1	1	2	2	1	2	4
1	1	2	2	1	2	4
2	1	2	2	2	2	3
2	1	2	3	2	2	3
2	2	2	3	2	2	3
2	2	2	3	4	2	3
2	2	2	2	2	2	3
2	2	2	2	2	2	3
1	2	2	2	1	2	4
1	1	2	2	1	2	4
1	1	2	2	1	2	4
1	1	1	2	1	2	4
1	1	2	2	2	2	4
1	1	2	2	2	2	4
2	1	2	3	1	2	2
2	2	2	3	3	2	2
3	2	2	3	1	2	2
3	2	2	4	2	2	1
3	2	2	4	2	2	1
3	2	2	4	2	2	1
3	3	2	4	1	2	3
2	3	2	2	2	2	3
2	2	2	2	2	2	3
2	2	2	2	2	2	3
1	1	1	2	4	1	4
1	1	1	2	4	1	4
1	1	1	2	4	1	4
1	1	1	2	4	1	4
1	1	2	2	1	2	4
2	1	2	2	2	2	3
1	1	2	2	1	2	4
1	1	2	2	1	2	4

2	1	2	2	2	2	3
2	1	2	3	2	2	3
2	2	2	3	2	2	3
2	2	2	3	4	2	3
2	2	2	2	2	2	3
2	2	2	2	2	2	3
1	2	2	2	1	2	4
1	1	2	2	1	2	4
1	1	2	2	1	2	4
1	1	1	2	1	2	4
1	1	2	2	2	2	4

Taking data from [1](see table 1) and using the algorithm from [3] we get a full cover of the data table by conjunctive sentences:

1.2 & 3.2 & 6.2 & 7.3 = 15 cases (the value of the first, third and sixth parameters is two and the value of the seventh parameter is three).

1.1 & 3.2 & 4.2 & 6.2 & 7.4 = 14

1.1 & 2.1 & 3.1 & 4.2 & 5.4 & 6.1& 7.4 = 4

1.1 & 2.1 & 3.1 & 4.2 & 5.1 & 6.2 & 7.4 = 2

1.3 & 3.2 & 4.4 & 5.2 & 6.2 & 7.1 = 3

Two objects are alone and are of no statistical interest

1.3 & 2.3 & 3.2 & 4.4 & 5.1 & 6.2 & 7.3 = 1

1.2 & 2.1 & 3.2 & 4.2 & 5.1 & 6.2 & 7.4 = 1.

An interesting and complicated case of conjunctive sentences is given by a triad

1.2 & 3.2 & 4.3 & 6.2 & 7.2 = 2

3.2 & 4.3 & 5.1 & 6.2 & 7.2 = 2

2.2 & 3.2 & 4.3 & 6.2 & 7.2 = 2

Those three pairs (seemingly 6 objects) are in reality creating a cover only for 3 objects.

All conjunctions in that list give us a full and non-doubling cover for the data table.

Therefore, having 1-to-1 correspondence between all condition sets and action sets we can reverse the covering sentences and get the so-called conditional productions [1]

1. IF (1.2 & 3.2) OR (1.3 & 2.3 & 3.2 & 4.4 & 5.1)
THEN (6.2 & 7.3)
2. IF (1.1 & 3.2 & 4.2) OR (1.1 & 2.1 & 3.1 & 4.2 & &5.1) OR (1.2 & 2.1 & 3.2 & 4.2 & 5.1)
THEN (6.2 & 7.4)
3. IF (1.1 & 2.1 & 3.1 & 4.2 & 5.4) THEN (6.1 & 7.4)
4. IF (1.3 & 3.2 & 4.4 & 5.2) THEN (6.2 & 7.1)
5. IF (1.2 & 3.2 & 4.3) OR (3.2 & 4.3 & 5.1) OR (2.2 & 3.2 & 4.3) THEN (6.2 & 7.2)

We have to mention that the author of [1] did not use the rule No 3 at all, although its covering value in the data table is 9 %.

So we have built up an alternative expert system for data table 1 from [1].

Creation of expert systems, using the method of monotonic systems gives us some interesting possibilities.

First, its speed allows us easily to implement new knowledge and practical follow-up results into an expert system. One can even talk about self-instructing expert system. Why? All rules will almost automatically be generated from a given data base. If new data are added, the represented method works directly on the same base and creates new conditional productions.

To characterise the speed of our algorithms we generated all conditional productions for a discrete 3000*7 data table (the first six parameters had 8 categories, the last one (0,1)). It takes only three minutes on IBM PS/2-50 personal computer.

References

1. Mrozek A. Rough Sets and Dependency Analysis among Attributes in Computer Implementations of Expert's Inference Models // Int. Man-Machine Studies. 1989. No 30. P. 457-473.
2. Vyhandu L., Mullat J. Monotonic Systems in Scene Analysis // Symposium Mathematical Processing of Cartographic Data. Tallinn, 1979. P. 63-66.
3. Kuusik R. Application of Theory of Monotonic Systems for Decision Trees Generation // Trans. of Tallinn Tech. Univ. 1990. No 705. P. 15-28.

L. Vyhandu, R. Kuusik

EXPERT SYSTEM AS A PART OF INFORMATION SYSTEM

Abstract

Expert system development, based on the theory of monotonic systems is described in this paper. Automatic generation of facts and rules for an expert system with a full coverage of a data table is estimated.

L. Võhandu, R. Kuusik

EKSPERTSÜSTEEM KUI INFOSÜSTEEMI OSA

Kokkuvõte

Artiklis kirjeldatakse montoonsete süsteemide teooria rakendust ekspertsüsteemides. Reaalsel andmetabelil esitatakse ekspertsüsteemi reeglistiku genereerimise protsess. Näidatakse, et seejuures jääb jõusse infosüsteemide üks põhinõudeid - kirjeldada uuritav andmekogum vähima arvu parameetritega.

No 723

TALLINNA TEHNIKAŰLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.06

L. Vyhandu

BEST ORDERINGS IN TOURNAMENTS

1. Introduction

Expert systems are very fashionable to-day. One can find heaps of articles on how to describe the behaviour of a real system with the help of conditional IF ... THEN rules. Just to recall that there exists a parallel and very powerful line of expert-style work. We will take up the expert estimation theory in the classical sense.

If one has to order a group of objects in some not strictly quantified sense, then the best way to do it is to gather a group of so-called experts and let them do the ordering work. That is an estimation task typical for beauty competitions, ski jumping, gymnastics. The same expertise is called upon if one has to choose the implementation order in a complex development project with contradicting requirements.

Typical expert estimation results in a data table. In our example there are data about 13 experts showing the relative importance of 6 problems. (We use a table from [1] as an example.)

Experts	Problems					
	1	2	3	4	5	6
1	1	4	3	2	6	5
2	1	4	3	2	6	5
3	2	1	3	4	5	6
4	2	4	5	1	6	3
5	4	3	5	2	6	1
6	4	2	3	1	6	5
7	5	4	6	1	3	2
8	6	5	3	1	4	2

9	6	5	4	2	3	1
10	1	3	4	2	6	5
11	1	4	3	2	5	6
12	4	1	3	2	6	5
13	4	3	5	1	6	2
Sum of ranks	41	43	50	23	68	48
Order of problems	2	3	5	1	6	4

It is easy to get the order of problems using sums of ranks given by experts:

4 - 1 - 2 - 6 - 3 - 5.

The same style is used in sports, where the sum of rankings given by judges defines the order of competitors.

Tournaments

To study the order of problems in a stable way we suggest another way. First, we find for all problems pairs i, j counts of how many times i is better than j and vice versa (i, j=1,...,6). The results of those countings are collected into a table (e.g., the ranks of problem 1 are 5 times better and 8 times worse than those of problem 2).

	1	2	3	4	5	6
1	*	5	9	5	10	8
2	8	*	8	2	10	7
3	4	5	*	1	11	7
4	8	11	12	*	13	11
5	3	3	2	0	*	2
6	5	6	6	2	11	*

(3)

As a next step we compare counts of all pairs aij and aji and make the following transformations:

If aij < aji then aij=0, aji=1,
if aij > aji then aij=1, aji=0,
if aij = aji then aij=aji=0.5.

After that transformation our table is as follows:

	1	2	3	4	5	6	
1	*	0	1	0	1	1	3
2	1	*	1	0	1	1	4
3	0	0	*	0	1	1	2
4	1	1	1	*	1	1	5
5	0	0	0	0	*	0	0
6	0	0	0	0	1	*	1

(4)

We have transformed our expert estimation problem to a tournament problem (there are no draws because of the odd number of experts).

Taking into account the points won by the problems they are ranked as follows:

$$4 - 2 - 1 - 3 - 6 - 5. \qquad (5)$$

Comparing the new order of problems with the direct ranking order (2) we can see that the first and the second problems have changed places, the same goes for the third and the sixth problem.

We have to ask a principal question: Which order is better?

To get at least a partial answer to that essential question we will use the connection between orderings in tournaments and Kemeny-Snell medians in nonparametric statistics.

We go again back to table (3) and arrange that table into the order (5) we got from the tournament table.

	4	2	1	3	6	5
4	*	11	8	12	11	13
2	2	*	8	8	7	10
1	5	5	*	9	8	10
3	1	5	4	*	7	10
6	2	6	5	6	*	11
5	0	3	3	2	2	*

(6)

It is easy to see that in this table the sum of elements above the main diagonal is maximal possible. The order of problems which guarantees such a maximal sum of elements above the main diagonal is called the Kemeny-Snell median and is generally estimated as the best stable ordering of elements in the space of permutations of expert estimations [2]. All exact algorithms to solve such problems are combinatorial. Usually one uses the so-called branch-and-bound method and the solution time grows approximately as $\sqrt{n!}$ if there are n objects to order [3].

So at the first glance it seems that we have got a very convenient and quick way of finding Kemeny-Snell median for expert estimates. We have to confess that the problem in our representation is oversimplified. In real situations we have more complex relations between possible orders, but all those can be brought down to a tournament table exactly as we did.

For a tournament table we can define the best order of participants as such which has a minimal amount of violations (that means - when a weaker player wins over a stronger one). For the general case of tournament we have developed a very effective greedy method as a mixture of the theory of monotonic systems and of the divide-and-conquer principle [4].

To demonstrate a more general case of solving expert estimation problems we analyze another example from [1]. Using the data about 9 teachers and their estimations by a large student group we get stable rankings and a tournament table:

	1	2	3	4	5	6	7	8	9
1	*	4	3	2	2	4	1	3	4
2	3	*	4	2	2	5	2	3	4
3	4	4	*	3	3	3	3	3	4
4	5	6	5	*	5	5	5	4	6
5	5	5	4	3	*	5	3	4	6
6	4	3	4	2	1	*	1	3	4
7	5	5	3	2	3	4	*	6	7
8	4	5	5	4	3	5	2	*	6
9	2	4	3	0	2	2	0	2	*

(7)

Bringing that table down to (1,. 5,0)-table we get:

	1	2	3	4	5	6	7	8	9		
1	*	1	0	0	0	5	0	0	1	2.5	
2	0	*	.5	0	0	1	0	0	5	2.0	
3	1	.5	*	0	0	0	5	0	1	3.0	
4	1	1	1	*	1	1	1	.5	1	7.5	(8)
5	1	1	1	0	*	1	5	1	1	6.5	
6	.5	0	1	0	0	*	0	0	1	2.5	
7	1	1	.5	0	.5	1	*	1	1	6.0	
8	1	1	1	.5	0	1	0	*	1	5.5	
9	0	.5	0	0	0	0	0	0	*	0.5	

Using the method described in [4], we eliminate player no. 9 as the weakest (having no wins). After that step every player has at least 1 win so we have to use divide-and-conquer principle. We take out all weaker players (that have at least as many losses as wins):

	To whom he lost	Wins
1	3	1
2	1	1
3	6	1
6	2	1

As one can easily see we have a full cycle of wins and losses. So we have to use an external rule to choose the weakest player. We take the player with the lowest sum of points (player no. 2) as the weakest. After that we get the order for weaker players: 9 2 1 3 6.

The stronger ones can be arranged into a table:

4	-	2
5	4	1
7	4	1
8	5 7	0

and the elimination order will be: 8 7 5 4. As the players 5 and 7 have at once no wins after eliminating the player no. 8, so we use an external rule: player no. 7 has fewer points and goes first.

After rearranging the tournament table we get:

	4	5	7	8	6	3	1	2	9		
4	*	1	1	.5	1	1	1	1	1	7.5	
5	0	*	.5	1	1	1	1	1	1	6.5	
7	0	.5	*	1	1	.5	1	1	1	6.0	
8	.5	0	0	*	1	1	1	1	1	5.5	
6	0	0	0	0	*	1	.5	0	1	2.5	(9)
3	0	0	.5	0	0	*	1	.5	1	3.0	
1	0	0	0	0	.5	0	*	1	1	2.5	
2	0	0	0	0	1	.5	0	*	.5	2.0	
9	0	0	0	0	0	0	0	.5	*	0.5	

As we can see there is only one violation in that ranking (player no. 2 won against no. 6). That is, however, the theoretical minimum of violations for that tournament (there is a cycle 6 - 3 - 1 - 2 - 6). The best ordering in that cycle is not so easy to find.

References

1. Beselev, S.D., Gurwich, F.G. Mathematical and Statistical Methods for Expert Estimation. Moscow, 1980 (in Russian).
2. Kemeny, J., Snell, J. Cybernetical Modelling. Moscow, 1972 (in Russian).
3. Flueck, J., Korsh, J. A Branch Search Algorithm for Maximum Likelihood Paired Comparison Ranking // Biometrika, 1974, v. 61, 621-626.
4. Vyhandu, L. Fast Methods in Exploratory Data Analysis. // Trans. of Tallinn Tech. Univ. 1989. No 705. P. 3-13.

L. Vyhandu

BEST ORDERINGS IN TOURNAMENTS

Abstract

How to find the best order of competing projects using expert estimation?

This article discusses some practical problems connected with the ordering of projects.

L. Võhandu

TURNIIRI PARIM JÄRJESTUS

Kokkuvõte

Kuidas leida eksperthinnangute abil võistlevate projektide parimat järjestust?

Artiklis käsitletakse projektide järjestamisega seotud praktilisi probleeme.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.3.06

F. Stuchlik
Sektion Informatik, Technische Universität "Otto von Guericke" D - 3040 Magdeburg, Boleslaw-Bierut-Platz 5

J. Tepandi

HUMAN FACTORS IN EXPERT SYSTEM DEVELOPMENT

Introduction

There is usually no sense to build an expert system (ES) solving a well-structured, well understood and established problem. For such problems, other types of software are less expensive. Expert systems may be cost-effective for ill-structured, non-formalised, evolving tasks that require judgmental and analytical skills. Development and delivery of such systems assume regular and high-intensity interaction between the users and the development team, therefore the role of human factors in ES development is very important. We may compare in this respect the traditional program and the ES life cycles. Firstly, there are many similarities: for example, the recognition of the business needs, selection of an appropriate task, and higher management support are crucial factors in any software system's success. Secondly, let us observe some differences. For the traditional programs, the importance of human factors (e.g. communication between the user and the developer) is very high in the early and final stages of development - during feasibility study, system analysis, integral testing and implementation. The design, coding and module testing phases are based on specification and therefore the impact of human-human communication is of less importance. In ES development the role of specifications is not very clear because of the ill-structured and evolving nature of the task. Therefore the role of human factors remains substantial all over the ES life cycle. We shall concentrate on issues specific to ES, omitting the requirements to computers and software common to any information system.

In particular, the ES validation process requires elaboration of human success criteria and is based on regular interaction between the developers and the users. In (Hayes-Roth, 1989) the success criteria are classified into task productivity and engineering quality criteria. The task productivity criteria include primary - advice quality, reasoning correctness, robustness, and solution efficiency -, and secondary criteria - cost, maintainability, ease of use, performance. The engineering quality criteria include primary - elegance of knowledge base design, modularity, architecture -, and secondary criteria - prototype support, graphics support, database support. The rest of the paper adds general remarks about human factors in ES development, concerns the interests of the ES users, gives an overview of ES validation and focuses on two specific studies. The first study deals with the task productivity validation and characterizes the tasks of the key roles in the ES validation. The second addresses the engineering quality criteria and investigates the optimum structure of human concepts for ES knowledge base.

Human Factors in Expert Systems: General Remarks

The expert systems and their development process may be considered from the different viewpoints. In particular, the users, the knowledge engineers, the artificial intelligence gurus, and the MIS department officials would give different answers to the questions concerning the role of human factors in ES development. We shall focus on the user's and the knowledge engineer's points of view.

Firstly, let us ask what does the user receive from an expert system in comparison with "traditional" software: data base systems, accounting software, programs performing numeric calculations, and so on. An expert system must provide expert level solutions to important problems, but this is not sufficient. Like an expert, an ES must be able to explain its reasoning and justify its conclusions. Furthermore, the problems solved by an ES should use heuristic knowledge, and the ES should be able to describe, manipulate, and update different types of knowledge: conceptual, structural, procedural, and factual. In contrast, the data base systems allow for easy manipulation of factual knowledge, but updating the conceptual and structural knowledge embodied in the data base schema requires restructuring of the database. That is very expensive, and therefore is performed very rarely. The application programs written in high-level languages usually make use mostly of the procedural knowledge, and so on. Therefore one of the most important characteristics of expert systems is their ability to utilize more types of knowledge than is used in conventional programs.

This leads us to the question whether these types of knowledge exhaust all ways of human thinking, i.e. whether ES represent human thinking. There have been different answers to this question, but by now it is clear that ES even do not reason like humans do. Moreover, there are features of human brain - take imagination and intuition, for example - that are not understood yet. Of course, they are not implemented in ES. On this ground, there have been talks about the ES crisis. We share the pragmatic view on the ES development, focussing on the results of the application, not on the ES/human similarities. From this point of view, the following comparison can be made. The cars do not have legs like humans, yet this is not the reason to speak about the crisis in the car industry - although, there may be crises for other reasons. Therefore, as far as ES efficiently solve important tasks, there is no sense to speak about their crisis.

Finally, we consider the following question, which is important from the human factors point of view: are ES autonomous or human-oriented? Traditionally, ES are seen as highly interactive. This view is supported by the ES transparency requirement (ES must be able to explain their reasoning). However, we must keep in mind the differences between ES shells and ES themselves; as well as the corresponding difference between ES development and its routine use. The ES shells have so far indeed been highly interactive. The situation may change with maturation of the automated knowledge acquisition tools, but this is not likely to occur in the near future. Also, the ES validation process seems to require human involvement for stating the goals and criteria of validation, as well as for evaluating the results. This situation will not change until ES are created by and for the humans. On the other hand, the ES itself may for some time function without human intervention, for example, in an autonomous mobile robot. So it can be said, that while in general ES require greater level of human/computer interaction than traditonal software, there exist both extremes: embedded ES that may work autonomously, as well as highly interactive medical and legal ES. Of course, the modification of an autonomous ES will again be the task of a human specialist.

What Are Expert System Users Really Interested in?

During the second session of the Society for Management of Artificial Intelligence Resources and Technology in Financial Services in New York, the results of a survey about the most important artificial intelligence topics were announced (Szuprowicz, 1988). These results indicate the following ten issues that are of major concern to expert system end users.

The utmost concern to them was connectivity of ES with the existing information processing environments. This is understandable if we look at enormous resources already put into hardware, software and corporate databases. The next concern was that of tools. Currently there are dozens of ES shells available in the market. End users do not have time and background to select the one that best fits their application.

The third issue mentioned most often was assessment of strategic artificial intelligence/ES applications. Indeed, if a task can be solved using conventional software, then the ES solution is usually more expensive.

Hence, it is important to find profitable applications that require ES methods.

Cost justification of introducing an expert system was the next significant issue raised. By their nature, the tasks solved by ES are complicated and the ES benchmarks are not established yet (Hayes-Roth, 1989). Also, the enterprises do not share willingly their experience in the competitive area of ES applications.

The following topics also ranked high among user interests: the process of introducing ES into an organization, use of ES for enhancing existing information processing applications, use of distributed ES, use of ES in conjunction with existing financial systems, computer-aided software engineering, and the relationship between artificial intelligence and MIS organizations.

In conclusion, the ES development seems mostly a responsibility of highly technical personnel who often lack the end-user orientation and business experience. This situation is similar to that in conventional programming in the near past, the changes in the situation may be expected in the near future.

The Key Roles in ES Validation

The goal of validation is to check whether the system under validation meets its initial objectives (have we built a right system?). This contrasts with verification - proving that the requirements of a previous development phase are correctly translated into methods and/or programs of the next stage. For example, one may verify that the system design corresponds to its specification. Verification may also involve proving that the knowledge base satisfies some well-designed formal criteria, such as consistency or completeness (Gaschnig and colleagues, 1989; EWICS TC7, 1986; Green and Keyes, 1987; O'Keefe, Balzi and Smith, 1987; Tepandi, 1989).

The validation process includes planning, testing, collecting the test results, and evaluation. During the planning phase, the following problems must be solved: How to specify the expert level? What and when to validate? Who must validate? What kind of tests will be performed? How to maintain the test data and collect the results? How high is an acceptable validation cost? The testing phase involves decisions concerning the test selection criteria,the procedures and persons responsible for elaborating the right answers for the test cases, the methods for avoiding systematical deviations in expert data and for dealing with controversial decisions of multiple experts. The results of testing must be collected and analysed in the last two phases of validation.

There are several groups of methods that can be used for ES validation:

- manual methods for early detection of errors in the knowledge base. Some examples: code inspection, manual tracing, scenarios of ES/user interactions;
- expert methods for checking the solutions proposed by the ES. Some examples: discussion of the test solutions with the experts and the users, a "gentlemen's collection" of tests for the ES domain, comparison of ES results with expert solutions, Turing tests, sensitivity analysis, visual analysis of ES/user interaction etc.;
- "white box" methods for the selection of tests based on the text of the knowledge base, e.g. rules, objects etc.;
- "black box" methods for test generation based on the problem description, not on the program text;
- different testing activities, such as random testing, subsystem testing, integration testing, field testing;
- statistical methods for the analysis of the results of validation, predicting the ES error rate and evaluation of the system.

In (Barker and O'Connor, 1989) the following key roles in the ES development are considered: the champion, the sponsor, the program manager, the technical team, the experts, and the users. The technical team includes the knowledge engineer(s) and the system integration engineer(s). Most of these roles are based on the belief in the system and its possibilities, and therefore, on constructive, synthetical and optimistic mindset. To successfully validate the system and commit to its success in real in-site environment it is useful to have also a person with the "doubting Thomas" type, destructive, analysing attitude towards the ES. This attitude may characterise some of the system developers, but also "a third party" - person/organisation who carries out validation. For example, Barker and O'Connor (1989) report that the quality assurance functional subgroup includes 10 percent of the members of Digital's Configuration Systems Development group.

In the following we will investigate the tasks and responsibilities of the key roles mentioned above in ES validation. Also, we try to predict who of the key players may have the necessary mindset for validation.

The champion is enthusiastic about the system and its future. Although (s)he may recognize the importance of validation, it is probable that the champion will tend to stress the strong points and ignore the weaknesses of the system. The role of the champion in validation is to recognize the importance of validation, to help in planning of the validation process, and not to insist on implementing an incorrect version of the system. The champion, as well as all other members of the development team, must be aware of the iterating, changing nature of ES building life cycle. This implies understanding that the insufficient quality of the first versions of the system is not an indication of the project failure, but a normal phenomenon in a prototype-based development.

The sponsor is interested in a real solution of the problem. (S)he has no time to take part in the development and should only insist that the validation activities be carried out properly.

The program manager is responsible for planning and supervising the validation process, and for interfaces between the members of the development team. (S)he is responsible for the validation as a whole, but does not carry it out.

The remaining key roles are playing an active part in all phases of the validation process. We will mention only some specific functions. The knowledge engineer is responsible for the knowledge base verification and the "white box" test selection. (S)he uses various interviewing techniques to establish a "gentlemen's collection" of tests for the domain.

The software systems integration engineer carries out subsystem testing and organizes the integration and field testing. The experts provide tests based on the problem to be solved and evaluate the reasoning, the conclusions and the explanation capabilities of the ES. The users must evaluate the functionality, quality, and user-friendliness of the system. It is also important that the users foresee the changes in the organization and in their jobs implied by the system, so that the potential effects of the project could be evaluated.

The Optimum Structure of Concepts in the ES Knowledge Base

As Hayes-Roth (1989) points out, the elegance of the knowledge base design, modularity, and interfaces are the most important intrinsic factors affecting the knowledge system quality. They foster incremental accretion of knowledge, localization of key concerns, and effective interaction with the system, as well as among the subsystems. There are many ways to achieve these goals. We have investigated one of them in (Tepandi, 1987), by carrying out an experiment concerning the optimum structure of concepts in the ES knowledge base. Below we will briefly characterize the motivation, the underlying model, the method, and the results of this study.

The motivation for the experiment was the belief that the structure of human concepts, the result of evolution for thousands of years, is (1) optimum for understanding the world, interaction with it, and changing it, and (2) optimum for the human-computer interface. So, understanding the characteristic features of this structure should be advantageous for any computer system design. In particular, this study is expected to be useful for expert systems and for object-oriented programming.

The model underlying our study is that of concepts having slots that specify relationships to other concepts (e.g., Lenat, 1983; Roussopoulos, 1979; Tepandi, 1987). In other terminology, the concepts may be frames or object classes, the slots - roles or attributes.

Our method was based on an assumption that explanatory dictionaries give the most understandable and comfortable definitions for the human users. So we used the Oxford (1980) and Russian (1957) as our sources of concept definitions. Twenty concepts were arbitrarily chosen from both dictionaries. For every concept, we examined its is-a concept (if there was one), as well as other related concepts. For all these concepts, the frame representations were created serving as a basis for investigating the relationships between the concepts.

The results of this study indicate that the new concepts are most frequently formed by adding a new slot with a new value to the is-a concept. Less frequently, an empty slot of the is-a concept is filled in. Replacing a value, heuristics (e.g. "make parts coincide"), and definition without an is-a concept are rarely used. In most definitions, the examples play an important role, illustrating the new concept or (not so often) defining a new value or a fundamental concept. This study indicated also, that the is-a hierarchies are not sufficient to represent the human concept structure - for this purpose, the is-a graphs seem to be more appropriate. The frequency of using different concept formation methods in both languages were rather similar thus allowing to propose the universal character of the concept formation mechanisms in these languages.

Acknowledgements

The authors are grateful to Prof. Bazewicz for helpful comments. The authors wish to thank A. Ida and I. Sinivee for their help in preparing the manuscript. This work was supported in part by Tallinn Technical University research contract OX-9022.

References

1. Barker V.E. and O'Connor D.E. Expert Systems for Configuration at Digital: XCON and Beyond. // CACM, 1989. 32, No. 3. P. 298-317.
2. Hayes-Roth F. Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering // Knowledge and Data Engineering. 1989. 1, No. 1. P. 101-110.
3. Gaschnig J., Klahr P., Pople H., Shortliffe E., and Terry A. Evaluation of Expert Systems: Issues and Case Studies // F. Hayes-Roth, D.A. Waterman, and D.B. Lenat (Eds.). Building Expert Systems. Reading, MA: Addison-Wesley, 1983.
4. EWICS TC7. Techniques for Verification and Validation of Safety-Related Software // Computers & Standards. 1986. 4, No. 2. P. 101-112.
5. Green C.J.R., and Keyes M.M. Verification and Validation of Expert Systems // Proc. Western Conf. on Expert Systems. IEEE Computer Society Press, 1987. P. 38-43.
6. O'Keefe R.M., Balzi O., and Smith E. Validating Expert System Performance // IEEE Expert. 1987. 2, No. 4. P. 81-90.
7. Lenat D.B. Theory Formation by Heuristic Search // Artificial Intelligence. 1983. 21. P. 31-59.
8. Oxford Advanced Learner's Dictionary of Current English. Oxford University Press, 1980. 1036 p.
9. Roussopoulus N. CSDL: A Conceptual Schema Definition Language for the Design of Data Base Applications // IEEE Transactions on Software Engineering. 1979. SE-5, No. 5. P. 481-496.
10. Russian Dictionary. In 4 volumes. / A.P. Yevgenyeva (Ed.). Moscow: State Publishing House of Foreign and National Languages Dictionaries, 1957-1961. Vol. 1, 963 p., Vol. 2, 1013 p., Vol. 3, 991 p., Vol. 4, 1080.
11. Szuprowicz B.O. What Are All Users Really Interested in? // Irish Computer, March 1988, p. 42.

12. Tepandi J. Concept Definition for an Information System End-User // Proc. Tallinn Tech. Univ. 1987. No. 645. P. 3-16 (in Russian).

13. Tepandi J. The Generation of Test Data for Rule-Based Expert Systems // Proc. 10th Int. Symp. Algorithms '89, April 17-21, 1989. Vysoke Tatry. Bratislava. P. 310-312.

F. Stuchlik, J. Tepandi

HUMAN FACTORS IN EXPERT SYSTEM DEVELOPMENT

Abstract

Expert systems may be cost-effective for ill-structured, non-formalized, evolving tasks that require judgemental and analytical skills. Development and delivery of such systems assumes regular and high-intensity interaction between the users and the development team, therefore the role of human factors in expert systems development is very important. We give general remarks about human factors in expert system development, characterise the interests of the expert system users, give an overview of the expert systems validation, investigate the tasks of the key roles in the expert system validation and study the optimum structure of human concepts for expert system knowledge bases.

F. Stuchlik, J. Tepandi

EKSPERTSÜSTEEMIDE LOOMISE ORGANISATSIOONILISI PROBLEEME

Kokkuvõte

Ekspertsüsteemid (ES) tasuvad end halvasti struktureeritud, formaliseerimata, muutuvate ülesannete lahendamisel. Selliste süsteemide loomisel on paratamatu tavalisest aktiivsem suhtlemine süsteemi loojate ja kasutajate vahel, seega on organisatsioonilised probleemid siin väga olulised. Arutame nende probleemide osatähtsust ES loomisel, iseloomustame ES kasutajate tegelikke huve, analüüsime võtmerolle ES valideerimisel ja uurime mõistete optimaalset struktuuri ES teadmusbaasides.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.3.06

P. Parmakson

TEST GENERATION AND KNOWLEDGE BASE BROWSING FACILITIES IN EXPERT SYSTEM SHELL HELI

1. Effective Test Generation for Expert Systems

Knowledge base testing can not be exhaustive, and one has to decide, when to stop it. Several criteria have been developed for this purpose. For example, according to the statement adequacy criterion, each rule must be executed at least once. This criterion is justified for expert systems (ES) without approximate reasoning: usually, when a rule is activated with all its premises satisfied, then it supports a certain conclusion and takes an active part in a decision process. In contrast to it, in approximate reasoning the decision is often evaluated as aggregation of all information available about it. Therefore, a rule may be activated with the given test data, but its contribution to the reasoning process may be negligible. In (Tepandi, 1989) a contribution adequacy criterion has been proposed, based on the following idea of efficient testing:

Each component of the rule base should be used effectively, i.e. its contribution should have a crucial impact on the inference process.

In (Tepandi 1989, 1990) several classes of ES with polynomial complexity and test data generation algorithms are characterized. In general, these problems are NP-complete. In this paper test generation and visualization methods in expert system shell HELI are presented and analyzed.

2. HELI - an Expert System Shell

The Heli expert system shell (Tepandi, 1988) uses a model of a rule-based expert system which resembles EQUANT (Hajek, 1985). The most important features of Heli are: rules, mixed control strategy, explanation facilities, the uncertainty handling method based on weights, actions associated with attribute-value pairs, syntax-directed rule base editor, graphic rule base browser, environment for test generation, ready-made demos, learning system, demo subsystem: possibility to store and repeat user actions, possibility to translate the system into different languages, working environment on IBM PC. HELI was developed by an AI Group of Tallinn **Technical University in 1987-1990.**

Knowledge base browser/tester (KBBT) is a subsystem of expert system shell HELI (Parmakson, Tepandi 1990). KBBT enables a knowledge engineer to: 1) display a part or the whole of the knowledge base (KB) graphically, as AND/OR-graph; to move the cursor around the KB graph and to study the completeness and consistency of KB visually, 2) generate effective tests for single rules or for entire KB according to a contribution adequacy criterion.

3. Basic Notations of Knowledge Base

Syntax of the knowledge base, simplified for this paper, has the following structure:

Knowledge base = {Rule}.

Rule = Antecedent "->" Conclusion.

Antecedent = Literal {"&" Literal}.

Conclusion = Literal.

Literal = ["not"] Proposition.

More frequently used are rule representation **languages** where a rule consists of a left hand side in a conjunctive normal form (or's of and's), and the right hand side is a conjunction of literals. One can transform all those rules into the previous format.

Let r be a rule, KB a knowledge base. Letters are used **below** to mark the propositions.

Goal is any proposition that appears somewhere on the right hand side of a rule and is "visible" to a knowledge base user, i.e. is meaningful to him. Propositions occurring only in the antecedents of rules are findings ("input"

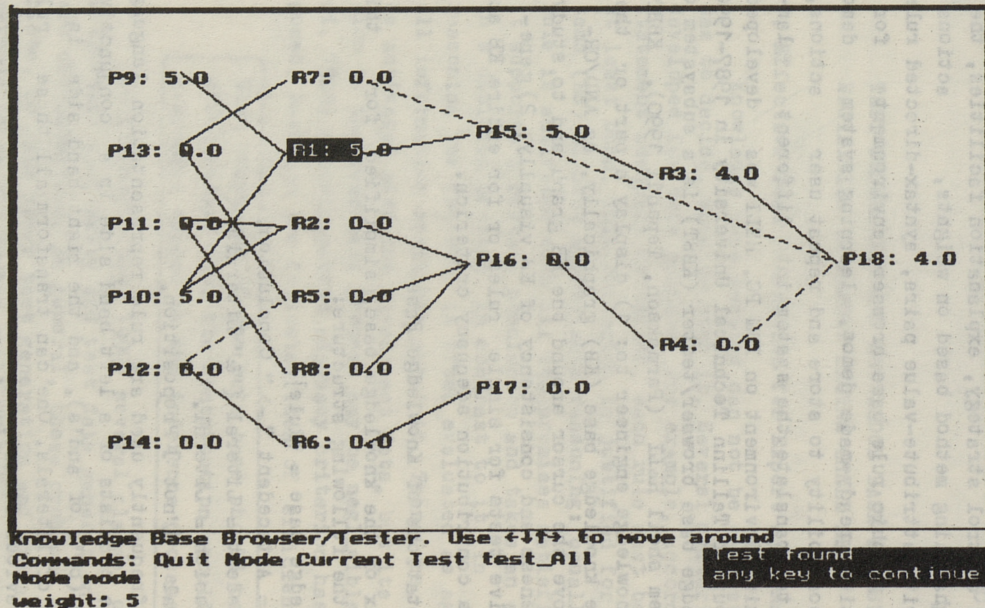

P9: 5.0
P13: 0.0
P11: 0.0
P10: 5.0
P12: 0.0
P14: 0.0
R7: 0.0
R1: 5.0
R2: 0.0
R5: 0.0
R8: 0.0
R6: 0.0
P15: 5.0
P16: 0.0
P17: 0.0
R3: 4.0
R4: 0.0
P18: 4.0
Knowledge Base Browser/Tester. Use ←↓↑→ to move around
Commands: Quit Mode Current Test test_All
Node mode
weight: 5
Test found
any key to continue

Fig. 1.

to inference). Propositions which are not findings are hypotheses. Let PR, G, F, H be the sets of all propositions used in the knowledge base, goals, findings and hypotheses respectively. F and H are complementary subsets of PR. G is a subset of H.

In Figure 1 there is an example of interactive session with KBBT. KB is loaded into KBBT with the following text:

r_1: p_9, p_{10} -> p_{15} weight 5

r_2: p_{11}, p_{10} -> p_{16} weight 5

r_3: p_{15} -> p_{18} weight 5

r_4: p_{16} -> p_{18} weight -4

r_5: p_{13}, not p_{12} -> p_{16} weight 3

r_6: p_{12} -> p_{17} weight 4

r_7: p_{13} -> p_{18} weight -5

r_8: p_{11} -> p_{16} weight 5

goal p_{18}

Rule r_1 can be read as follows: IF p_9 and p_{10} THEN p_{15} (with certainty factor 5). Negative weight of the rule is equivalent to negation of conclusion of the rule, e.g. rule r_4 can be written otherwise as

p_{16} -> not p_{18}.

4. Graph Representation of KB

To obtain a clear understanding of the structure of a knowledge base it is expedient to represent the knowledge base as a AND/OR graph (term 'knowledge network' is also used). By organizing rules and propositions into a graph, the search for matching rules can be avoided.

To construct the graph of knowledge base one must

- put the propositions and rules used in a knowledge base into one-one correspondence with proposition nodes of the graph. However, if we broaden the knowledge base syntax to allow disjunction in antecedent of rules, then the rules in a knowledge base must be split into several rule nodes;

- if proposition p occurs in antecedent of rule r, then there is an edge from node p to node r. We assume that proposition can occur in antecedent of the rule only once. If the proposition in the antecedent is negated, then the edge is dashed;

- if proposition p occurs in a conclusion of rule r, then there is an edge from node r to node p. If the proposition in the conclusion is negated, then the edge is dashed.

Example 1. A knowledge base consisting of five rules is shown below:

r_1: P & Q -> A

r_2: Q & R -> B

r_3: S & not T -> C

r_4: A -> D

r_5: B -> not D

goals: C, D

The graph for this knowledge base is shown in Figure 2.

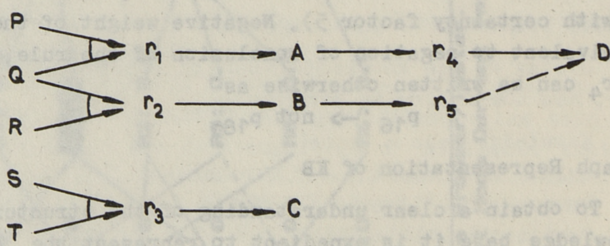


Fig. 2

In the example above G = {C, D}; F = {P, Q, R, S, T}; H = = {A, B, C, D}.

The inference in the rule-based system can be viewed as propagating truth values in the AND/OR graph beginning from the findings.

4. Partitioning Propositions and Rules into Levels

Assuming that rule base is acyclic, it is possible to partition its propositions and rules into levels (cf. Ginsburg,1987). KBBT divides the propositions and rules of KB into levels so that on the left side of the screen there are findings; goals are placed on the right. In locating the AND/OR-graph on the screen some esthetic criteria should be met, e.g. the length of the edges of the graph should be minimized. In case of large knowledge bases, one can slide the window and alter the scale.

5. Three-valued Inference Scheme

There is a large diversity of inference schemes used in knowledge bases. In accordance with the goal of system portability, KBBT may be adjusted by several inference schemes. At the moment there do exist methods of effective test generation for knowledge bases which are developed on the basis of two- and three-valued propositional logic and the inference scheme used by the system HELI. In Figure 1 an inference scheme is used where each rule has the weight in interval (-5,5), the values of propositions are in the same interval. Value 0 denotes semantically the propositions "don't know" or "not available".

We take an inference scheme from HELI shell as the basis and simplify it as follows: 1) a priori weights are null for all propositions; 2) the weight of the rule can be -1 or 1. Negative weight means that the conclusion is negated; 3) in contrast to the conventional logic, three truth values are used: 0, -1 and 1. Truth value -1 means "false"; truth value 0 means "unknown", "irrelevant"; truth value 1 means "true".

Advantage of the three-valued logic is that due to truth value 0 the result on the application of logical operations is always determined.Therefore the description of test generation algorithms is simpler. From the complexity point of view the three-valued logic is located between the propositional calculus and the inference scheme used in HELI shell.

In the course of inference the "logical operations" negation, conjunction, disjunction and modus ponens are used.

For three-valued logic these operations are established by truth tables shown below.

P	not P	mp(P)
-1	1	0
0	0	0
1	-1	1

P	Q	P or Q	P & Q
-1	-1	-1	-1
-1	0	-1	-1
-1	1	c	-1
0	-1	-1	-1
0	0	0	0
0	1	1	0
1	-1	c	-1
1	0	1	0
1	1	1	1

Algorithmically:

$P \& Q = \min (P, Q)$

$P \text{ or } Q := (P + Q) / (1 + P * Q).$

In case of operation 'or' the value pairs of P and Q (P = -1; Q = 1) and (P = 1; Q = -1) are of special meaning - then the result is considered to be contradictive.

The definitions of logical operations described above are not the single possible. The operation P or Q:= max (P, Q) has been used (e.g. Schmitt 1988), but we prefer a definition which assures the compatibility with HELI inference scheme.

6. Effective Tests

Test t is an evaluation of findings. Rule r fires when in the label of the rule the argument of mp operation has the truth value 1; in opposite cases the rule is not involved in the inference. Test t is effective for the rule r if in application to the KB and KB-r the values of at least one goal are different.

Let us examine effective tests for a knowledge base in the example 1. It is clear that the rule r_3 can be effectively tested here, so we concentrate upon the other rules. Suppose the ordering of findings is lexicographical, then the list (0, 1, -1) notates findings evaluation P = 0; R = -1. Findings evaluation (1, 1, 1) leads to contradiction.

There are two effective tests for rule r_4: (1, 1, 0) and (1, 1, -1), and also two effective tests for rule r_5: (0, 1, 1) and (-1, 1, 1).

What can be said about effective tests for rules r_1 and r_2? The test which is effective for rule r_4 is effective for rule r_1 as well. The same holds for rules r_5 and r_2.

7. Main Ideas for Effective Test Generation Algorithm

We shall not discuss the test generation algorithm here in detail. We follow only the moments which are important from the complexity point of view.

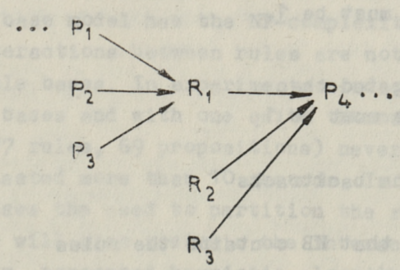
We can start from the rules at the lowest level, moving up until test generation for all rules is done. In prototype version one can point out the rule for which tests have to be generated. In addition one can generate tests for the whole KB and save the tests into testing log.

For each rule r all paths to goals will be generated. Using one such a path at a time the algorithm tries to find out an effective test.

If an effective test for a rule can be found, when considering some path, then this test is effective for other rules in the same path as well.

In the path containing effective tests all propositions must have different values for KB and KB-r. So in the path containing effective test all rules must fire.

Consider a fragment of a rule base:



We want to check if there is an effective test for the path ... P_1, R_1, P_4, ..., or not. Values of propositions P_1, P_2 and P_3 must be such that the rule R_1 fires. Rules R_2

and R_3 must not fire. Based on those conditions, one can derive values for propositions occurring in premises of rules R_2 and R_3, and also contributions from rules to propositions P_2 and P_3.

The main concern in an algorithm is AND- and OR-branching points. It is clear that a larger set of possible values means more choices at branching points. When propositions can have values from the contiguous interval, then this method does not work.

There are three types of branching points:

1. Suppose KB contains the rule

r_1: P_1 & P_2 ... P_n -> D

where P_1, P_2, ..., P_n, D are literals.

In the course of test generation the support of r_1 to D has already been determined. Now we must check: if propositions p_i have already values, then if they conflict with the value of support; if any propositions p are uninstantiated then we must give them values in an appropriate way. Here is generally more than one path to follow.

```
if D = 0 then
    at least one of P_i-s must be -1 or 0;  {branching point
                                              of type 1}
if D = 1 then
    if D is negated then
      algorithm backtracks
    else
      all P_i-s must be 1,
if D = -1 then
    if D is negated then
      all P_i-s must be 1
    else
      algorithm backtracks
```

2. Suppose that KB contains the rules

P_1 -> D
P_2 -> D
...
P_n -> D

where P_1, P_2, ..., P_n are antecedents and D is a conclusion.

In the course of test generation we have already chosen a value for D. Now we must check, if instantiated supports from premises already have values, then does these conflict with the value of D; if any supports from premises are uninstantiated then we must give them values in an appropriate way. There is generally more than one path to follow.

if D = 0 then

all supports must be 0;

if D = -1 then

at least one support must be -1 and others may be 0 or -1; {branching point of type 2}

if D = 1 then

at least one support must be 1 and others may be 0 or 1. {branching point of type 3}

Usage of negation does not influence complexity of the algorithm.

The used algorithm is based on relatively straightforward width-first search. In branching points mentioned above the algorithm tries several combinations. The algorithm memorizes branching points and backtracks, then reaching the dead end. One type of a dead end is contradiction of rules.

8. Is the Algorithm Feasible?

It has been shown that effective test generation for a used rule base model has the NP-complexity (Tepandi,1990). However, interactions between rules are not so intensive in practical rule bases. In experiments conducted with several sample rule bases and with one quite thoroughly engineered rule base (77 rules, 69 propositions) never has the test generation lasted more than 10 seconds. In case of large knowledge bases the need to partition the rules into modules arises. That will also restrain the interactions between rules. Anyway, presented heuristic algorithm is better than simple inspection of all combinations of finding evaluations.

9. Test Generation: Approximate Inference Scheme

We applied the algorithm discussed above to an approximate inference scheme too. The main difficulty here is the threshold - the difference of which size can be designated as substantial? For about 90 % of rules the tests found are appropriate for quite low thresholds. In addition the good test generation problem can be stated otherwise: find such a test that applied to KB and KB-r gives maximal difference in the values of the goal. The difference found characterises the participation of the rule.

In Figure 1 the knowledge engineer gives the command to generate effective test for rule r_1. KBBT displays the generated test, showing values for each finding as well as the firing of rules and values of hypotheses and goals. One can store the generated tests into the testing log.

The case of instantiation of propositions shows that the absence of rule r_1 leads to value of goal $p_{18} = 0$; in the presence of rule r_1 the value of goal p_{18} is 4.0, therefore rule r_1 is absolutely necessary.

There is no effective test for rule r_2. The reason is that when rule r_2 fires, then rule r_8 always fires as well. Rule r_2 is superfluous in the KB. Generally, one can say that when there is no effective test for a rule, then there is a redundancy in the KB. In the given example rules r_2 and r_8 have maximum weights; the situation changes when those weights are not maximal. Then we can state the problem of finding the effective test as follows: effective test is an instantiation of findings such that the contribution of rule into value of goal is the largest.

10. Interdependence of Test Generation, Rule Generation and Redundancy

Next we define the equation system representation of rule bases. Label is a function that sets to every evaluation of F into correspondence a truth value of proposition or rule.

- Label of the finding is the identificator of a finding;

- If the literal is negated, then the label of the literal is symbol "not" concatenated with the label of the literal.

- Label of the rule consists of labels of literals in the antecedent, separated by "&" and put between symbols "mp(" and ")". If the conclusion of the rule is negated, then in addition symbol "not" is put ahead of the label.

- Label of the hypothesis consists of labels of all rules where the hypothesis is in conclusion, separated by "or" symbols.

The rules and goals of knowledge base above are described with the following labels:

r_1: mp(P & Q)

r_2: mp(Q & R)

r_3: mp(S & not T)

C : (S & not T)

r_4: mp(P & Q)

r_5: not mp(Q & R)

D : (mp(P & Q) or not mp(Q & R))

So one can represent the rule base as a system of equations

$$g_1 = L_1(f_{11}, f_{12}, \ldots, f_{1,k})$$
$$g_2 = L_2(f_{21}, f_{22}, \ldots, f_{2,k})$$
$$g_n = L_n(f_{n1}, f_{n2}, \ldots, f_{n,k}),$$

where $g_i \in G$; $f_{ij} \in F$; and L_i is a label of g_i.

From here we can set up a problem opposite to a test generation: given are the examples about the behavior of the knowledge base to be constructed; generate the rules, i.e. the labels of the goals. Automatically generated rules can be hardly comprehensible to human user, but the automated rule weight adjusting would be very valuable. It seems that those contrary problems are also hard for computation.

Several definitions of a rule base redundancy have been proposed, e.g. - a rule base is redundant if it contains one or more rules which can be removed without changing the set of conclusions asserted by the rule base for any set of valid inputs (Ginsburg,1987). Using equational notation which has been taken into use above, one can define another redundancy criterion - the knowledge base is redundant if there exists another knowledge base with the equivalent system of shorter or simpler equations.

Definitions of the effective test and redundancy are quite close. To define redundancy we consider all conclusions, whereas to define an effective test only goals are considered. If all conclusions are goals, then finding an effective test for each rule is sufficient to prove the non-redundancy of the rule base.

11. Further Directions

Our experiences show that textual representation of KB is convenient in the first stages of KB development, but not sufficient in verification. In addition to AND/ /OR-graph KB can be represented as a decision table. At the present time the syntax-directed editor is used to edit the KB of HELI, but one of our further directions is to add to browser the facilities of KB editing to integrate KB testing and editing.

Inability to generate the effective test for a rule indicates redundancy, but it is not clear where the cause of redundancy is and what has to be done to avoid it. The relationship between test generation and redundancy is still not clear yet and requires further research.

References

1. Ginsberg A. A New Approach to Checking Knowledge Bases for Inconsistency and Redundancy // Proc. 3rd Annu. Expert Syst. Gov. Conf. Washington, D.C., Oct. 19-23, 1987. P. 102-111.
2. Hajek P. A Theory of Propagation of Uncertain Knowledge in Consulting Systems // P. Hajek (Ed.). New Enhancements in GUAH Software. Praha, 1984. P. 37-47.
3. Parmakson P., and Tepandi J. An Overview of Expert System Testing with Remarks on Distributed Architecture // Bazewicz M. (Ed.). Proc. XIth Scientific School on Distributed Information Systems Architecture. Wroclaw, 1990.
4. Schmitt P. Computational Aspects of Three-Valued Logic. Karlsruhe, 1988.
5. Tepandi J. HELI - an Expert System Shell // Trans. of Tallinn Tech. Univ. 1988. No. 671. P. 12-19 (in Russian).
6. Tepandi J. Test Generation of Expert Systems with Fuzzy Truth Values // Bulletin for Studies and Exchanges on Fuzziness and Its Applications. 1989. 39. P. 83-92.
7. Tepandi J. Verification, Testing and Validation of Rule-Based Expert Systems // Proc. of the IFAC 11th World Congress, Aug. 13-17, 1990. Vol. 7. P. 162-167.
8. Weber A. Eine Methode zur Aktualisierung Aussagenlogischer Wissensbasen. Diss. Univ. Fridericiana Karlsruhe, 1987.

P. Parmakson

TEST GENERATION AND KNOWLEDGE BASE BROWSING FACILITIES IN EXPERT SYSTEM SHELL HELI

Abstract

In this paper test generation and visualization facilities in expert system shell HELI are presented and analyzed. A short description of an effective test generation algorithm is given. Interdependencies of effective test generation and knowledge base redundancy checking are discussed.

P. Parmakson

TEKSTIDE GENEREERIMINE JA TEADMUSBAASI KUVAMISE VAHENDID INSTRUMENTAALSES EKSPERTSÜSTEEMIS 'HELI'

Kokkuvõte

Artiklis käsitletakse instrumentaalse ekspertsüsteemi HELI koosseisu kuuluvaid testide genereerimise ja teadmusbaasi kuvamise vahendeid. Esitatakse efektiivsete testide genereerimise algoritmi lühike kirjeldus. Arutatakse ka seoseid testide genereerimise ja liiasuse kontrollimise vahel.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.06

T. Luige

STRUCTURED DATA PROCESSING USING COMPILER WRITING SYSTEMS

1. Introduction

With the growth of database applications for information retrieval and processing systems data input is and will be of major concern. Such information as encyclopedia articles for knowledge bases, records describing words for dictionaries, bibliographic records for libraries, texts for photocomposer printing, etc. usually need specific preliminary work to render it understandable for the computer, i.e. to make the computer distinguish between different data elements and to link the needed actions to them. In many cases it is a most labour-consuming activity. Along with the development of optical reading facilities it would be convenient to handle a wholly automatic data input. It would minimize the cost, time and amount of monotonous human work if no data retyping or special worksheet completing is required.

New data input system development for non-trivial textual input forms involves consideration of every document processing system which is based upon a document model. The common model as a long string of characters is often sufficient, but the more demanding the applications are the more expressive models are required. In the above-mentioned cases the data elements are distinguished by their place in the record and their relations to other data elements rather than their exact beginning position or special marks or labels. Conventional paper documents can be broken down into structures of ordered objects (sequences). Thus the body of an article could consist of a sequence of sections, each section containing a sequence of paragraphs and each paragraph containing a sequence of atomic text blocks (sentences, enumerated items, strings, etc.).

To mark these cases let us introduce a notion of a structured object implying a document or a way of data presentation which is determined by its structure rather than a certain sequence, length and type of elements. The rules serving for compilation of those documents determine their basic structure but certain elements can be either present or absent, there may be a choice of elements, the elements or whole groups may be repeated.

On a certain level of abstraction compilation rules for a document can be regarded as a language. This approach is valid for every kind of documents being most fruitful for the structured objects and enabling us to describe them with the mathematical concepts used for programming languages. It means also that the automatic systems of textual input can be created using the means serving for automatic processing of programming languages. At the Department of Data Processing of TTU a compiler writing system ELMA has been worked out which is used as a generator for conventional compilers. It is also gaining importance as a generator of analysers for little languages and any non-trivial textual input forms.

An information system "Dictionary" has been worked out using these strategies. The dictionary articles are considered a language and are transformed into an inner format for the computer with the help of a compiler. The input for the dictionary system came from the tape that was first meant for a photocomposer and had some additional separators which made it possible to build an input generator. Optical reading facilities might even enable us to give an input straight from the pages of a ready dictionary. Sequentially given dictionaries are formally described and their correctness is tested. This formal description allows knowledge already present in the computer to be used in compiling new dictionaries, it also makes statistical as well as contents analysis easier.

The author has mostly dealt with the input system of bibliographic records that is further discussed in more detail. Both systems have proved to be quite successful or at least executable. However, the problems encountered are common to different processing systems of structured objects.

2. An outline of automatic bibliographic data input system

There are two common ways of data input: either to give certain space or certain separator for every element in the record. Still, if a record is very complicated these approaches prove inadequate. A bibliographic record is a good example: it has about 40 different elements whereas only a few are obligatory, others may be and very often are absent, all of them may be repeated, either separately or in groups. In most cases only 10 of the 40 elements possible are represented, the length of many elements is not restricted (e.g. some authors' names or headings may be very long). It is impossible to give every element a certain space. It is also cumbersome to give each element an individual label because a whole system of labels is needed for this purpose. A good deal of work is required and many errors occur. It would be most convenient to leave this kind of work to the computer.

Attempts have been made to automatically recognize bibliographic data elements earlier. Advanced computer technology facilitates compilation of the retrospective electronic catalogs even for the biggest libraries. If we assume that one bibliographic record consists of 700 marks on the average, a machine catalogue for 1 million bibliographical units will take about 700 Mbyte. Thus information about 10 million books can be stored on one set of magnetic disks [3].

This paper describes an attempt to design new user-friendly methodology for bibliographical data input, based on formal grammars and languages. A compiler writing system ELMA serves as a basis for programming, most importantly for transformation of retrospective catalogs into a machine-readable form. These entries exceed millions. The data about new books are presented on magnetic tapes in a standardized format.

Bibliographic descriptions are compiled according to a standard [6] that determines the sequence of data elements in the record and the separators between them. Information that can be used for distinguishing elements is:

1) order of elements;
2) punctuation marks (.-.,:;/ // ()[]+ -);
3) key-words;
4) different print and size of text;
5) beginning of a new line.

Bibliographical record is divided into an inscription and areas separated by ".-". In every area the elements have a certain sequence and are detached by certain separators determined by the standard. The basic structure of the bibliographic record can be described by the following scheme (Fig. 1):

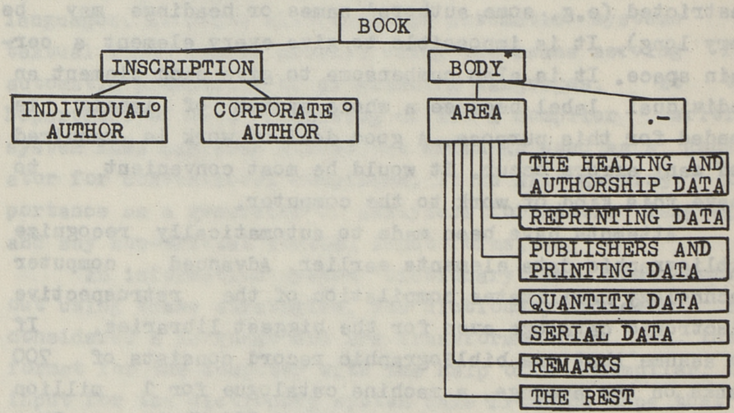


Fig. 1

All the areas consist of different elements, e.g. the heading and authorship data include all the headings (main, second, supplementary and parallel headings, names of all authors, including editors, illustrators, translators, etc.). Quantity data include the number of pages, illustrations, the format and the same data about appendixes. The area of remarks often contains the headings and authors of all component parts of the book. The last area includes the ISSN and ISBN registration numbers and the cost.

The scheme of one area might be as follows (Fig. 2):

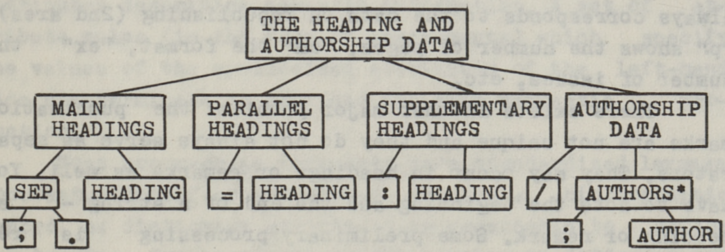
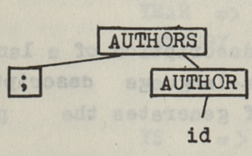
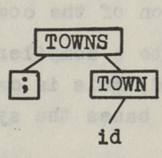


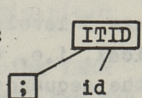
Fig. 2

The element in the record is determined both by its location and separator - mostly a punctuation mark. The separator solely does not identify the element. For instance ";" and "," are very often encountered and it leads to reduction conflicts in the grammar. In order to eliminate equal right sides of the rules you have to find the common part and introduce a new non-terminal. For instance instead of



and


you have to take a new production:



and replace it everywhere where such situation is encountered. Connecting attributes with these rules means that the two have to be parsed by common productions until the difference can be seen in the look-ahead and the values of synthesized attributes must be carried to that point. The above form of syntax is artificial. This kind of inconvenience is hard to accept as the problem would be solved easily by examining the information associated with the identifiers in the symbol table [4].

To make the recognition of the elements easier you can also use the key-words, i.e. words that identify the elements they are attached to. For instance, "Rev. ed." always corresponds to the data on republishing (2nd area), "p." shows the number of pages, "cm" the format, "ex" the number of issues, etc.

There exists another major problem: the punctuation marks are not unique and they do not always serve as separators. They may occur in headings or remarks as well. You have to note the beginning and the end of a string - a heading or remark. Some preliminary processing is required and this inconvenience is very hard to overcome. It might be possible in terms of a sequence and the obligatory elements. But it might not be reasonable because it may lead to many errors.

Using key-words to identify elements also raises the question of the languages applied in bibliographic records. Most of the literature in our libraries is in Estonian and Russian. We may also write the compilers for English, German and French books but in some cases we must do without any key-words.

3. Description of the compiler

To write a compiler a formal description of a language is needed. This is called a meta-language description and on these bases the system itself generates the processor [5].

There are 3 levels describing a language:

1) lexical, i.e. the main structures used, the rules of dividing the sequence of input symbols into logical parts - lexemes (e.g. word, punctuation mark);

2) syntactical level - gives the structure according to which the lexemes are grouped into a record;

3) semantical level - gives interpretation rules for structural components.

The description of a bibliographical record can also be given in terms of lexemes, structure (syntax) and semantics.

The semantic part of the metalanguage ELMA consists of the attributive formalism augmented with the use of external functions (written in FORTRAN). An attributive grammar consists of a context-free grammar in which non-terminals are associated with synthesised and inherited attributes describing semantic information. A set of attribute rules (in the form of assignments) which specify the values of the synthesised attributes of the left-hand side non-terminals is provided for each syntactic production.

Most programming languages have standardized lexemes; constant, identifier and string. But for a bibliographic record or dictionary article it is reasonable to define new lexemes. For instance, 'year' appears in different parts of the record. It can be described as a constant but there are several restrictions: the year has to begin with at least 17, 18, 19, or 20, there may also be a period of time that is marked by "-", thus the description of a lexeme TIME can be as follows:

```
TIME    => TIME , YEARS
        => YEARS
YEARS   => YEARS - YEAR
        => YEAR
YEAR    => CENTURY YS
CENTURY   =>  18
          =>  19
          =>  20
YS      => Y Y
Y       => 1|2|3|4|5|6|7|8|9|0
```

The drawback of simply declaring YEAR => const is that a constant in FORTRAN might end with the "." and there will be a reduction conflict if the "." is also declared as a separator. The other thing is that the analyser automatically gives an error if the year is impossible (publication in 1288 instead of 1988, etc.). When defining a lexeme, a list of possible values can be given.

Different analysers are needed but the main lexemes are alike. Thus it is reasonable to put the description of the lexemes apart. Defining TIME as a lexeme makes it easier to detect elements like "publishing time", "printing time" from other numbers of the record which usually go as a constant.

The standard describes 8 types of bibliographic records quite similar to each other (monographs, serials, collections, technical documents, non-published items, analytic records, etc.). In addition several different standards have been used throughout the years. The differences are not big: somewhere ";" used instead of ",", only "." is used instead of ".-" in older versions. These changes greatly influence the analyser because element differentation is based on these punctuation marks. There are 3 ways to overcome this obstacle:

1) to make the grammar universal, to take all of these variants into account;

2) to compile different analysers for different standards and make the computer to choose between them (this is used already for the language choice made by a specific compiler);

3) to handle these varieties of standards as errors and in the case of reduction conflict change it on the screen as if correcting errors.

Analysing all the standards and types of records we choose the way and time for use. It is reasonable to make a preprocessor which first of all scans the record and then decides which analyser to use. From among different bibliographical records about 2/3 may be processed using 7-8 analysers for different types. There always remains a certain percentage of bibliographic cards that will not fit into the scheme and need preliminary processing on the screen. It is difficult to say what the trade-off between automation and user involvement should be in the design of error handling. Evidently, if a high degree of automation is preferred, the quality of error-recovery should be as independent of the form of the grammar as possible. At the other extreme, if user involvement is encouraged, the syntactic metalanguage should be augmented with an additional high-level mechanism by which the compiler designer can easily specify the behaviour of error recovery. Evidently the compiler should show the place where something goes wrong. It is still to be decided whether it should alter the bibliographic record itself and then pass it again or leave it to be done by the user. With increased automation it would lead to difficulties because bibliographic records have a complicated structure and the elements are very similar to each other.

4. Data presentation

After a formal description of the bibliographical record a problem arises with regard to inner format of the records. MARC is a widely spread format for national standards for processing bibliographical and authorship data in interlibrary cooperation [1]. It involves a set of content designators prescribed for data identification and characterization in a particular category of machine-readable record. Every MARC record has a fixed-length data element made up of four character positions which describe the type of a record and bibliographical level (monograph, serial, collection and analytic). The rest of the record consists of fields of variable length, each containing one bibliographic element and having a three-number tag which identifies the element.

The common communications format used in the East-European countries is called MEKOF and is based on the same principles: a positional part of fixed length describing the record and the elements marked with a tag (a 6-position 16-system number) in fields of variable length.

The bibliographic record analyser and the inner format generator are independent of each other. This is important because there are several types of MARC formats and changes are made to establish a consistent one for any kind of bibliographic records. So the analyser must be changeable too. The next step is to make the inner format wholly independent of the work of the analyser and give the tags in the form of a tree-structured variable which is automatically matched to the description of the bibliographical record.

5. Summary

In the USSR Central Library investigations on the possibilities of automatic recognition of bibliographic data elements based on the established standard were carried out [3]. An experimental data base was made of 1000 records, analysed statistically and some recommendations were given to design a bibliographical data input system. It was discovered that such elements as the main heading, output data (time and place of publishing and printing) and qualitative characteristics (number of pages) were always present, more frequently also the cost (90.8 %) and notes (83.7 %). This information is valuable for compiler writing. Obviously, the more frequently the element occurs the higher is the effect of automatic recognition programs.

In practice a compiler for one type of bibliographic record consists of about 200 rules, it uses 11 separators and about 60 or more key-words. A considerable part of these belong to the authorship data and characterize author's relation to the book. If the inner format does not demand different tags for writers, illustrators, composers, etc., many of these may be omitted. When the system is running it takes only some seconds to analyse one record, it is convenient for the user to follow and the delay is not too long.

An automatic transformation from bibliographic record into machine-readable communication format has proved feasible. The whole system of bibliographic data input is a major concern. The question is: how to merge different types of bibliographic records into one formal description and to minimize the number of different analysers? The rest of the differences may be handled as errors and corrected on the screen.

Some areas requiring development are:

1) user interface design - on the whole the system is not oriented on the dialogue with the user. But the inconsistencies must be dealt with in the style and vocabulary of a bibliographer;

2) the system must be adaptive and capable of development, it must cope easily with altering situations and demands.

A general method for modeling and a representation scheme of non-trivial textual input forms have been developed. Although a rather complicated approach for document description is used instead of a natural and intuitive way a wide variety of document classes can be handled in a uniform and consistent manner. A specification of new documents and document classes is facilitated.

The validity of the approach is confirmed for purposes of organizing computer input of structured data which can be described as a tree-structured variable, identify the areas where this approach is appropriate and raise critical issues. Such a system has a much wider area of application than a library. It involves both automatic reading and recognition of elements of bibliographic records and can also act as an interface between a printed text and a computer data base. It is related to natural language input for computers and artificial intelligence.

References

1. Attig J.C. The concept of a MARC format // Information Technology and Libraries. March 1983. P. 7-17.
2. Kimura G.D. A system editor for abstract document objects// IEEE Transactions on Software Engineering. Vol. SE-12. 1986. No. 3. P. 417-435.
3. Kolotygina V. Possibilities of automatic transformation of the bibliographic records to the communication format// Trans. of State National Scientific and Technical Library. 1986. P. 108-115 (in Russian).
4. Koskimies K. et al. The design of a language processor generator// Software Practice and Experience. Vol. 18. 1988. No. 2. P. 107-135.
5. Lepp M., Vooglaid A., Vyhandu L. ELMA as an instrumental system for building practical programming systems// Trans. of Tallinn Tech. Univ. 1986. No. 614. P. 85-91.
6. System of Standards "Information, librarianship and publishing". Bibliographic description for a document. General requirements and rules. Moscow, 1984 (in Russian).

T. Luige

STRUCTURED DATA PROCESSING USING COMPILER WRITING SYSTEMS

Abstract

An input system of non-trivial textual forms is introduced. An input of bibliographic data elements and an automatic recognition system serve as an example. Formal grammars and languages are used and programming is based on a compiler writing system ELMA.

T. Luige

STRUKTUURSETE OBJEKTIDE TÖÖTLEMINE TRANSLAATORITE KOOSTAMISE SÜSTEEMIDE ABIL

Kokkuvõte

Artikkel käsitleb keerulise struktuuriga tekstide töötlemist arvutis. Taoliste tekstide koostamise eeskirjad on käsitletavad keelena ja järelikult võib neid töödelda samade vahenditega kui programmeerimiskeeli. Artiklis on välja toodud struktuursete objektide mõiste ja näitena antud ülevaade bibliokirjete sisestamise süsteemi realiseerimisest translaatorite koostamise süsteemi ELMA abil.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.3.016

T. Lumberg

ORGANISATIONAL PLANNING IN TERMS OF INFORMATION SYSTEM DESIGN

We know far more about making good practice
than we actually put into practice

Organisational planning refers to the need to locate any single development effort into the context of the organisation as a whole, and its future development. Increasing attention has been paid to this area. Some problems arise when the context is difficult to determine, as can be seen in our economy, due to high-level uncertainty.

The need for this level of planning may be associated with the complexity of the problems being tackled by an organisation. The boundaries of an information system are often fuzzy, and systems are as much social as technical. Without planning, the development of information systems will lead to systems which are not integrated with each other or the strategic needs of the organisation. When such planning is performed there is more likelihood of information systems tackling the right problems, at least in organisational terms. This in turn leads to more stability in the political environment during the life of a project (Downs, 1988).

The opportunities opened through technological developments are many, and some planning of these from the organisation's point of view may vastly increase the contribution of information systems to the institution's goals and missing opportunities. This implies that managers of an organisation have a policy or plan for the future which includes a strategy for employing information technology to support the organisation. Once the strategy for the use of information technology (IT) has been built around the organisation's strategy, plans may be developed to move from the existing to the ideal.

But high-level uncertainty is the feature of our present economy. This state of organisation's environment drives managers to develop new structural and performance solutions in the organisation itself. Actual environmental changes demand further analysis of organisation's structure and new-level definition of its strategy.

It is a complex task to develop a new strategy for an organisation in a framework of economic relations which has changed dramatically. Lack of experience and overwhelming uncertainty in economic trends seem make this development task impossible. We must start from determining the starting points. In the design of the system a designer needs some basis for modeling the organisation and the changing process to be implemented (Scott and Booker, 1979). Thus, we must have a good basis to define the starting points of strategical decisions. It is useful to think about the organisation in terms of major sets of interacting (interrelated) variables. For that purpose we will develop the conceptual framework of strategical decisions.

We attempt to achieve a dual goal.

1. To develop a good perception of an organisation by its functional orientation in the economic environment (goal-oriented view).

2. To describe and validate this perception by means of a formal model, which can serve as a basis for decisions and improvement actions in future. However, it would lead to information system design (informational view).

The set of actions of an organisation within internal and external environment forms a net of different interactions (transactions) and flows with the corresponding informational reflections and flows. Respectively, functioning of an organisation forms an information system and a communication net and criteria for the latter ones (comprehensible, complete, closely related, processual and flexible). On the other hand, these features enable in bottom-up approach to derive certain common rules for organisational planning on the basis of the so-called goal-oriented informational view.

As can be seen, we do not make sharp distinction between an organisation and its information system (IS), not even in the abstract sense. It is assumed that an information system, in the broadest sense of the word (including information, information work, information flows and the corresponding means), can be regarded as the skeleton of an organisation. So, organisation analysis can be regarded as IS definition and IS design as an organisation design. At this point we can point out that this is the only way of how to integrate IT into an organisation. (Kerr, 1988; Moad, 1989). The above-mentioned two views constitute a single systematic view, named as goal-oriented informational view, by which we aim at overall development of organisations as economic and information systems.

The development and implementation of the informational view is only one of the possible formalisations suitable to aid the so-called "decision track". Decision track here refers to the set of causally related strategic decisions, including goal-, function- and structure settings, technological decisions, which will design the organisation, i.e. affect its response function (Kornai, 1971). We assume these decisions to be causally related, because each concrete decision simultaneously defines the scope of the other ones, including alternatives and possible implementations.

Why do we suppose it to be best suited for such purpose? First, informational view is a comprehensive view. The complexity of the information structure coincides with the complexity of organisations and leads to the separation of functional units called upon to perform definite information and control activities within the organisation. Neglect of the functional division within the organisation leads to an oversimplified description of the motivations underlying behaviour, just as neglect of the complexity of the informational structure fails to do justice to reality (Kornai, 1971). Secondly, simultaneous strategical planning, description and optimisation of the organisation as a whole by decision track as much as its IS excludes some integration problems (Burbidge, 1987), gives us the model of effectively goal-oriented organisation (not by idealising, but optimising!) and well-tailored IS solution. If we can determine the scope of an economic system and subjects within it, and observe certain behavioural regularities, then we can design the structure of the goal-oriented economic organisation.

Organisational planning methodologies are orientated to a management perspective which is trying to use information technology to achieve the goals of senior staff. They are therefore inappropriate for situations where these goals are the subject of investigation. Involvement of goal-setting problems in our consideration demands also the analysis of the economical background / environment of an organisation. So we will further refer to János Kornai, who has investigated mathematical modeling of behavioural and decision processes under uncertainty on different levels of an economic system. Once behavioural regularities of an organisation on the economic system level are described and analysed by J. Kornai (stochastic-causal description of economic system), we may further deal with internal problems of an organisation on the above-mentioned basis.

We have found that terms (and means, respectively) of IS design as high-level concepts have common features, which enable us to use them as a basis and support in the analysis and strategy (re)creation of an organisation. We will relate these features with IS design concepts in the organisational context, thus building up the above-mentioned conceptual framework for strategic decisions. Description in terms of IS design involves presentation of concept-feature pairs given below. In practice this conceptual framework can be "filled up" with values about the organisation under investigation.

1st concept: ORGANISATION AND SCOPE

In this concept an organisation is regarded as a communication system closed on the users' (of the output) needs and wants. Charles R. Litecky has described it as follows: as an example, the first step is to assess the users' needs. Then product, processes and distribution methods are planned so as to meet these desires effectively and efficiently. The planning information flows to operations when the plans are carried out and the product (goods or service) is produced. The product is then promoted so that users will be aware of its existence and how it will serve their needs. Finally, the product is distributed to them, at which time the needs and desires of users are satisfied. This closed-loop communication system should operate continuously if the organisation is to be efficient. This serves as an illustration of how the design of a system must constantly rely on feedback (Litecky, 1981). As external users' needs change, the changes could be discovered as quickly as possible by management, plans revised, and operations changed accordingly to best serve the revised needs.

As can be seen, organisation's functioning area (i.e. communication area) is not restricted with its territorial or main function area. Indeed, here we have the area of organisation's interests as an economic subject. All these activities, objects, flows which are relevant to end result, constitute organisation's scope. If we (re)define the strategy and goal, then we will (re)define the scope.

Note. Here we avoid deliberately using the term "structure", as by goal-oriented approach the forming of the organisation's concrete structure is relatively flexible and not long-life, depending on forming and action of goal-oriented units.

2nd concept: STRATEGY AND GOAL

The following contradiction emerges here. On the one hand: the way we think today is not suitable after 3-4 years! On the other hand: strategical change is never completed in a moment! It shows the importance of treating strategy as a process. In general we can regard strategy as a choice of possibilities (trends) for increasing organisation's competition power. In its best strategy is focused on the goal.

In system design strategic requirements need to be formulated, as they indicate the objectives and goals expected to be accomplished by the organisation. Else there may be system failures that could have otherwise been avoided. Strategic requirements simply involve task requirements. They may be general or broad objectives or more specific. Once the strategic issues are evident, information systems can be designed to receive full benefits.

Goal is a two-level term: the overall goal of the organisation can be regarded as one's mission, which defines what the organization wants to be in terms of its activity area; the more detailed goal derives from an earlier one and shows the specific targets to be sought at specified points in time. Regarding the strategy concept, we are also looking at the goal having both meanings in mind.

To achieve the objectives and accomplish the mission, a company must set substrategies, for example, the market, the product and the manufacturing strategy. The manufacturing strategy must be connected to the market and product strategies - it must be derived from the latter and at the same time it must support and influence them. The latter must themselves be closely coupled and their choice must be the result of a careful analysis of the external environment, the company's objectives, and the restrictions (resources, weaknesses) of the company itself (Ang, 1989).

By establishing the implications of the mission on the organisation's function, action objectives can be set and prioritised. They are set in terms of critical factors (i.e. factors which contribute to the organisation's overall success). The objectives should then translate into a strategy, which defines in broad terms what changes must take place within organisation's scope to achieve these objectives.

3rd concept: PROCESS AND FUNCTION

Before we can even start to consider system development, we should understand the entire processing spectrum and its short-and long-range plans, and document an understanding of what presently takes place and what will take place (Davis, 1981).

The goal of processes in an organisation is to enable it to produce end-result, that meets one's objectives. Often there is a clear perception about evident, main processes in the organisation (e.g. manufacturing → technology, equipment, materials, personnel, which are temporally and spatially tied by events). But that evidence is in a certain sense a conservative factor, too. Some processes may be covert and unclear (informal relations, real action chains, information flows, decision-making, etc.), which however have a great impact on the end-result.At the same time, the organisation strives for a better result. Entire development effort in the organisation can be treated as a process.

Goal-oriented, flexible and effective process management is possible if the processes can be controlled,measured and improved. Intrinsic to that concept 'process' is the feature 'function' - process fulfil function. We can define, control, measure and improve a process via functionality or functional parameters by means of qualified information, which determine the process maturity level. The perception of the latter is, in its turn, the background of development process. Tasks can be defined, subdivided into manageable parts and then the key variables are identified and controlled.

The following discussion concerning the process is derived from (Humphrey, 1989).

A process is said to be stable and under statistical control if it is predictable within certain established limits. When a process is under statistical control, repeating the work in roughly the same way will produce roughly the same result. So, to get consistently better results, it is necessary to improve the process. If the process is not under statistical control, sustained progress is not possible. That process consists of tools, methods, and practices that result in the desired product. To be fully effective, the process must take into consideration the relationships between all of them, as well as the skill, training, and motivation of the people involved.

4th concept: SUBJECT AND ROLE

Organisations as much as information systems exist in a social context because they are developed by, interact with, and their existence is justified by people. Attention to this aspect is an important focus for the attention of our view. By (Downs, 1988) two examples can be presented on how to treat the development process in socio-political terms. Firstly, the various 'parties' try to manipulate the components of a specific project to influence the design so that it meets their own goals. Secondly, as we see organisations as dynamic, continously developing and socially constructed through the choices and initiatives of individuals, systems may be considered to emerge from this turbulent social interaction. This 'processual perspective' may lead to a redefinition of the roles of participants in development projects (i.e. subjects), and the creation of more explicit structures to support and benefit from these social and political interactions.

With special attention on that trend which is carefully expressed here, and is of major importance under uncertainty, we are implementing the logic of goal-oriented management in our consideration.

Subjects may be described as rational functional objects with needs and facilities, thus with goals and a desire to achieve them. Such subjects range from states and organisations to groups and individuals. There is the informal and frequently covert aspect of these subjects, their social face, including culture and motivation. So it is wrong to adopt a functional view of people as 'human factors'.

If a subject acts in order to achieve one's goal, it is in interaction with other subjects and an effective interaction assumes combining different individual goals into a certain common goal and the corresponding harmonised stable behaviour. In other words, that interaction forms the role of a subject. Such consideration coincides with the commonly known definition of a role as a socially expected behaviour pattern, usually determined by an individual's status in a particular society.

CONCLUSION

These concepts as abstractions from entities in real world exist in mutual cohesion, which become evident by relationships of named features of these entities:

1) goals(s) of a strategy in an organisation is (are) determined by subject(s);

2) scope of the organisation is determined by goal(s);

3) function(s) of processes in an organisation is (are) determined by goal(s) and scope;

4) role(s) of a subject in organisation is (are) determined by goal(s), scope and function(s).

Here concepts perform the stable forms of the corresponding entities in real world and features perform one's dynamic values. Thus, the conceptual framework leads us to a conceptual model, which includes entities and relations as abstractions, enable us to reflect and model in a common (universal) formalised form static and dynamic characteristics of the organisation on a convenient logical level, while the conceptual framework enables many-levelled interpretation.

Specification of these concepts and the corresponding relations on different decision levels (an organisation specific decomposition) in the form of models of goal-oriented informational view ensures simultaneous and smooth planning and development of the organisation itself and its information system. The purpose is: to substitute the processual stability of the organisation for the structural one in order to decrease organisation's response time and increase flexibility in a rapidly changing economical environment. For that purpose we have aimed at defining a framework that will help organisations establish priorities for improvements.

References

1. Ang C.L. Planning and Implementing Computer Integrated Manufacturing // Computers in Industry. 1989. No. 12. P. 131-140.

2. Burbidge J.L., Falster P.P., Riis J.O., Svendsen O.M. Integration in Manufacturing // Computers in Industry. 1987. No. 9. P. 297-305.

3. Davis, Keagle W. The Information Systems Auditor of the 1980's // Management Accounting. March 1981. P. 40-47.

4. Downs Ed., Clare P., and Coe I. Structured Systems Analysis and Design Method. Application and Context. Prentice Hall International (UK) Ltd. 1988. P. 217.

5. Humphrey W.S. Improving the Software Development Process // Datamation. April 1, 1989. P. 28-31.

6. Kerr S. IS Trims Down // Datamation. 1988. Vol. 34, No. 19. P. 46-49.

7. Kornai J. Anti-Equilibrium. On the Economic System Theory and the Tasks of Research. North Holland Publishing Company, 1971. P. 390.

8. Litecky C.R. Corporate Strategy and MIS Planning // Journal of Systems Management. January 1981. P. 36-39.

9. Moad J. Navigating Cross-Functional IS waters // Datamation. March 1, 1989. P. 73-75.

10. Scott R.L. and Booker J.A. A Three-Phased Approach to the Systems Development Audit // Internal Auditor. June 1979. P. 26-31.

T. Lumberg

ORGANISATIONAL PLANNING IN TERMS OF INFORMATION SYSTEM DESIGN

Abstract

Organisational planning refers to the need to locate any single development effort into the context of the organisation as a whole, and its future development. Once the strategy for the use of IT has been built around the organisation's strategy, plans may be developed to move from the existing to the ideal.

The article deals with development of conceptual framework of strategical decisions. For that purpose IS design concepts are used in order to derive certain common rules for organisational planning on the basis of the so-called goal-oriented informational view.

T. Lumberg

ORGANISATSIOONI PLANEERIMISEST INFOSÜSTEEMI PROJEKTEERIMISE TERMINITES

Kokkuvõte

Organisatsiooni planeerimine on seotud vajadusega vaadelda iga arendustegevust organisatsiooni üldises kontekstis. Kui infotehnoloogia rakendamise strateegia on kujundatud vastavalt organisatsiooni strateegiale, võib välja töötada plaanid üleminekuks olemasolevalt ideaalsele.

Käesoleva artikli teemaks on strateegiliste otsuste kontseptuaalse aluse väljatöötamine. Selleks kasutatakse infosüsteemi projekteerimisest tuntud kontsepte, tuletamaks reegleid organisatsiooni planeerimiseks tulemusorienteeritud infovaatest lähtudes.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 386

L. Elmik

AN OFFICE AS A COMPLICATED SYSTEM

1. Introduction

Office modelling constitutes one of the fast growing area of Information Systems [1, 6, 8]. Many research efforts have recently been conducted for deriving systematic guidelines and theoretical foundations for the design of Office Information Systems (OIS) able to support information management in the office environment [1, 2, 3, 4, 5, 7, 9, 10, 14].

In developing a methodological tool to guide OIS conceptual design, a fundamental element is the type of office conceptual view that is adopted during the analysis of the office. Just as in conventional information systems design, several types of conceptual views of the reality of the enterprise can be considered, and different views result in different approaches to the analysis of office work. Office conceptual views can be classified as technical, organizational and socio-technical.

A technical view examines office work in great detail, considering the basic operations performed in the office. An organizational approach based on the global organizational structure of an office is considered; the hierarchical structure of the organization is thus reflected in the office model. A socio-technical view looks at the office in terms of the tasks to be performed by each unit into which the enterprise is subdivided.

In this paper the author will present a communicative approach to the office conceptual modelling in which a fundamental element is supporting management with information.

The communicative approach can be classified as socio-technical, but in contrast to other approaches it looks at the office as a whole rather than at separate units.

2. Structure of a management process

The theory of management states that management is rather an art than science: because managers use all their skill and knowledge independent of know-how sources [11, 12]. It is also emphasized that the goal of any management and control process is to decrease chaos and there exists only one way to do this: to create information of this area [13].

Without information available it is not feasible to organize management of an enterprise. At the same time OIS cannot only be treated in terms of the tasks to be performed by each unit: because a task can be changed, and in the real world they do this.

2.1. Presentation of a management process

A management process involves environment subjects (or objects) and relations between them.

Formally:

$$M = f(E, O, R)$$

or

$$M = f(E, S, R),$$

where

E - environment;

O - object;

S - subject;

R - relation.

Some examples on management:

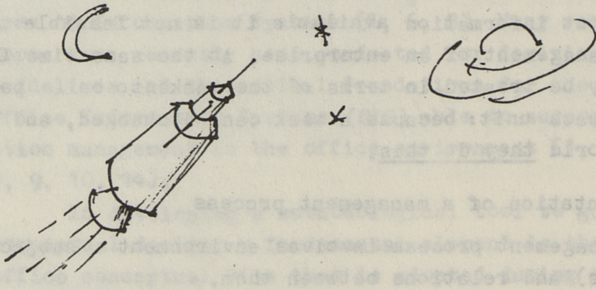
1.

A man manages the behavior of a stone, and as the result of his action the stone changes its place in environment.

2.

A ballerina manages her behavior, and as the result changes her positions and place in the environment.

3.



A rocket installed with navigation instruments is managed by a pilot who changes the behavior of the rocket in environment.

There are many approaches to understand the relationship between environment and subject (object) [3, 5, 14].

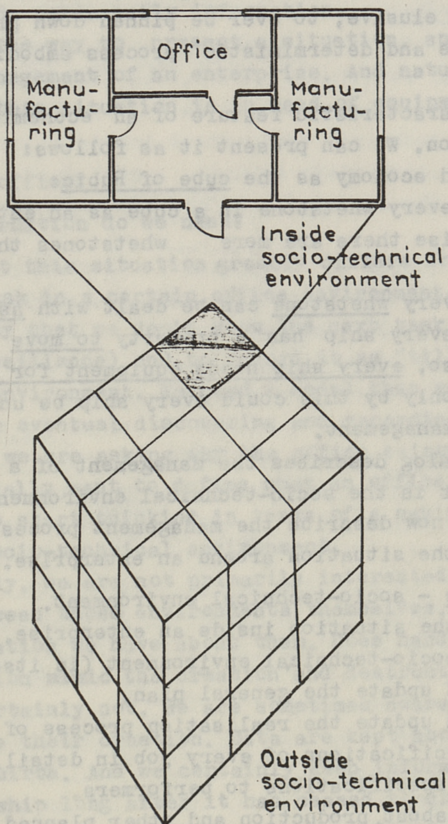
A useful way for programmers is management described as an algorithm (step by step), for example [14]:

1) estimate a situation;
2) design a general plan;
3) design the realization process of the plan;
4) design specifications of every job in detail;
5) transport specifications to performers.

An algorithm and relationships described thus can be used for controlling production and its changes. But for OIS modelling it is not adequate.

This structure does not provide for the following:

1) what information is needed;



Model of relationships between two socio-technical environments

2) how to handle information.

Management of an enterprise in its "real" essence is probably too amorphous, too ambiguous, too subjective, too slippery and elusive, to ever be pinned down precisely by the objective and deterministic process embodied in a computer.

The characteristic feature of an economic environment is integration. We can present it as follows:

- world economy as the cube of Rubic:
- and every whetstone in a cube as an enterprise (in a real enterprise there are more whetstones than in the cube);
- and every whetstone can be dealt with as a ship;
- and every ship has a property to move (to sale);
- and so, every ship needs equipment for navigation:
- and only by this could every ship be under the control and management.

The analog describes the management of a socio-technical subject in the socio-technical environment.

We can now describe the management process as follows.

1. Estimate the situation around an enterprise.

   Outside - socio-technical environment.

2. Estimate the situation inside an enterprise.

   Inside - socio-technical environment (in itself).

3. Design and update the general plan.
4. Design and update the realisation process of the plan.
5. Design specifications of every job in detail.
6. Transport specifications to performers.
7. Feed-back about production and other planned processes.

The result is an interface - between the planned and the real.

The algorithm presents the structure of a self-organising or cybernetic management in economy.

In this algorithm management means "to bring into order" upon the two estimates: the outside and the inside ones, and after that to take up interior activities inside an enterprise.

These activities are effected by the two information flows, which collaborate and oppose simultaneously. Incidentally, we must not forget, in the situation of acting inside that socio-technical environment in economy changes every moment and depends on this increasing chaos and an enterprise needs permanently information.

This is one way to present a situation, and thus to realise the management of an enterprise. And naturally every enterprise in this situation is in need of equipment of navigation.

3. What is an office?

3.1. What information do we need?

Note that this situation greatly differs from a simple design of a task in a certain office environment. It is not uncommon to say that we don't know the next task (for example: pollution surveillance) and to record it as "till now unknown" in an office environment. We need a model that would include facilities for eventual discovering and recording (what it is). Instead, we are asking why the office exists at all.

If we really want to define what an office is, we should have to start thinking in terms of a navigation process into a socio-technical environment.

Certainly, we are not primarily interested in the relations between those environments themselves, but rather in the information we have about them. Does handling of this information mimic the creation and destruction of such relations? Certainly not. We are sometimes aware of the relations before their creation. Data are kept about children before their birth. And we certainly keep information about some relationship long after it has ceased to exist.

What is the place where we are to collect information about a relationship between the two environments?

Of course, it is an office.

But, is an office a place?

If we are interested in the information, it is not only a place.

It is rather a system, the adapters of which exist everywhere: inside and outside an enterprise. Largely it is a special kind of an artificial intellect.

Now, we can fix ranges of the concept of an "enterprise". We regard an "enterprise" as a synonym to the term "unit to manufacture, sale and purchase something". And we shall deal only with the information that is considered as a reality to decrease chaos by manufacturing, sale and purchase processes.

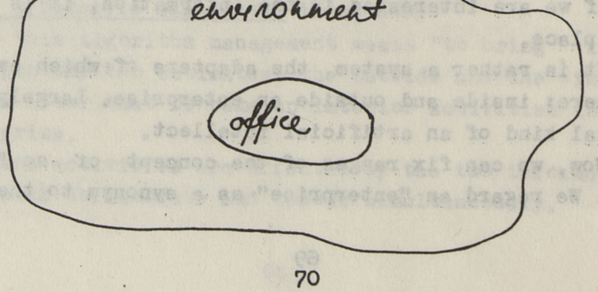
It is clear that within an enterprise we have more detailed and structured information (less chaos). This information to a large extent is to depend on us. It means that socio-technical environment inside an enterprise should come under appropriate control. A good deal of information handled by the structures is created according to the needs of managers.

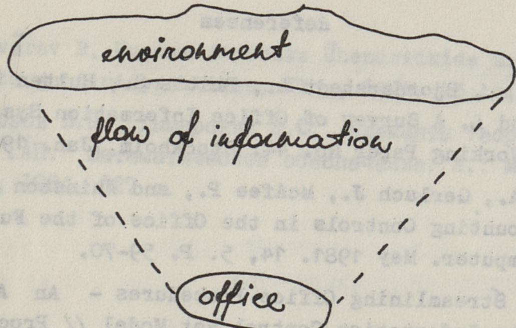
But outside an enterprise it is more undetermined. In the whole economic environment where a certain enterprise is only one of the numerous factors acts on them. The action of an enterprise on the environment does not usually give as important information as all factors.

We have reached a situation where we have poorer instruments in an office to obtain important and complicated information from outside than we have to handle information about inside. By designing only the existing tasks in an office environment, we never come closer to the solution of the problem: how to navigate an enterprise better, how to create and use more useful information, how to add (to use together) outside and inside information flow.

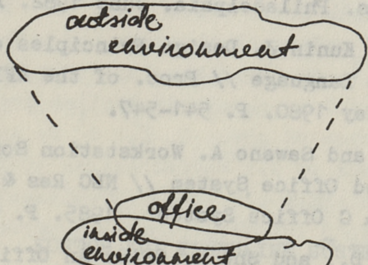
To show the place of an office we can present the situation as a picture.

The first view does not distinguish between outside and inside information. And thus we will be unable to construct a model of an office, involving the structure of a self-organised management, because undetermined and controlled environments are not distinguished.





In the second case, we can think about an office in the context of a self-organised management, because we do distinguish an active management area from a passive one and determine an inside environment from an undetermined outside environment.



Thus we have presented two approaches, the first is a traditional one, the second model serves for presentation of an office and supporting the description of a self-organised management structure.

The second model points out the following:

An office is a complicated system [14]. A system is complicated if it consists of at least one natural intellect (one human), and aims to support the management process of an enterprise by processed information and information processing, to connect, harmonise, bring together, coordinate an undeterminated informations flows.

References

1. Ahlse'n M., Bjornerstedt A., Britts S., Hulten C., and Soderlund L. A Survey of Office Information Systems // SYSLAB Working Paper No. 44. Stockholm, Jan. 1983.

2. Bailey A., Gerlach J., McAfee P., and Whinston A. Internal Accounting Controls in the Office of the Future // IEEE Computer. May 1981. 14, 5. P. 59-70.

3. Cook C. Streamlining Office Procedures - An Analysis Using the Information Control Net Model // Proc. of the AFIPS National Computer Conference. May 1980. P. 555-565.

4. Ellis C. and Nutt G. Office Information Systems and Computer Science // Computing Surveys. March 1980. 12,1. P. 27-60.

5. Gibbs S. Office Information Models and the Representation of 'Office Objects' // Proc. of the ACM SIGOA Conf. on Office Systems. Philadelphia. June 1982. P. 21-26.

6. Hammer M. and Kunin J. Design Principles of an Office Specification Language // Proc. of the AFIPS National Computer Conf. May 1980. P. 541-547.

7. Kobayashi K. and Sawano A. Workstation Software System for Integrated Office System // NEC Res & Develop. Special Issue on 'C & C Office System'. 1985. P. 97-101.

8. Lum V., Choy D., and Shu N. OPAS: An Office Procedure Automation System // IMB System Journal. 1982. 21, 3. P. 327-350.

9. Zisman M. Use of Production Systems for Modeling Asynchronous, Concurrent Processes // Waterman and Hayes-Roth, eds. Pattern Directed Inference Systems. New York: Academic Press, 1978. P. 53-68.

10. Zloof M. QBE/OBE: A Language for Office and Business Automation // IEEE Computer. May 1980. 14, 5. P. 13-22.

11. Üksvärav R. Majandusorganisatsioonide juhtimine Ameerika Ühendriikides. Tln.: Eesti Raamat, 1966. 104 lk.

12. Üksvärav R. Organisatsioon ja juhtimine sotsialistlikus majanduses. Tln.: Valgus, 1974. 304 lk.

13. Üksvärav R. Uuemat Ameerika Ühendriikide majandusettevõtte juhtimispraktikas. Tln., 1974. 43 lk.

14. Морозов В.П., Дымарский Я.С. Элементы теории управления ГАП. Математическое обеспечение. Л.: Машиностроение, 1984. 327 с.

L. Elmik

AN OFFICE AS A COMPLICATED SYSTEM

Abstract

A self-organised management structure and a communicative navigation model of an enterprise and an office is defined.

L. Elmik

KONTOR KUI KEERULINE SÜSTEEM

Kokkuvõte

Antud artiklis on käsitletud kontori kui infosüsteemi modelleerimise probleeme. Esitatud on erinevad suhestumismudelid juhtimisprotsessis ja juhtimisprotsessi struktuur küberneetilistes süsteemides. Töö põhiresultaat on kontori definitsioon ja tema eesmärkide määratlemine.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 519.4

P. Puusemp

REDUCED ABELIAN GROUPS AND THEIR SEMIGROUPS OF ENDOMORPHISMS

1. Introduction

It is well known that the isomorphism of two periodic Abelian groups G and H follows from the isomorphism of semigroups of all endomorphisms of G and H ([1], theorem). In this paper we shall study a similar problem for torsion subgroups and semigroups of endomorphisms of a reduced Abelian groups. We shall use the methods and ideas from [1].

For a suitable group G let $\pi(G)$ be a set of all prime divisors of orders of elements of finite order of group G. Our aim is to prove the following theorem.

THEOREM. Let G be a reduced Abelian group and T(G) the torsion subgroup of group G. Let G^* be a group such that the semigroups of all endomorphisms of groups G and G^* are isomorphic. Then all the $\pi(G)$-elements of G^* belong to the center $Z(G^*)$ of G^* and form a subgroup isomorphic to T(G).

From this theorem it directly follows

COROLLARY. Suppose that G and G^* are two reduced Abelian groups such that the semigroups of all endomorphisms of G and G^* are isomorphic. Then the torsion subgroups T(G) and $T(G^*)$ of G and G^* are also isomorphic.

We will use the following notations: Z(G) - a center of a group G; End G - a semigroup of all endomorphisms of a group G; I(G) - a set of all idempotents of End G; C_t - a cyclic group of order t; $\langle a \rangle$ - a subgroup, generated by a. For Abelian groups we use instead of term "direct product" a term "direct sum".

2. Preliminaries

Let G be a suitable group and x, y $\in$ End G. We say that x and y are summable if gh = hg for all g $\in$ Imx and h $\in$ Imy, i.e. a map x + y, defined by g(x+y) = (gx)(gy), is also a endomorphism of G. We say that x and y are orthogonal if xy = yx = 0. This is based on [2], p. 303.

LEMMA 1 ([3], lemma 1.1). If x $\in$ I(G), then group G splits up in the following semidirect product:

$$G = \mathrm{Kerx}\,\lambda\,\mathrm{Imx}.$$

Let x, y $\in$ I(G) and z be a homomorphism from Imx into Imy. In view of lemma 1, the homomorphism z can be uniquely extended to an endomorphism of G by setting (Kerx)z= $\langle 1 \rangle$. In this paper we identify homomorphism z everywhere with its extension on G.

LEMMA 2 ([1], lemma 1). If x,y $\in$ I(G), then Hom (Imx, Imy) = $\{z \in \mathrm{End}\ g \mid xz = z = zy\}$.

LEMMA 3 ([4], lemma 4). If $x_1,\dots,x_n \in$ I(G) and the idempotents $x_1,\dots,x_n$ are pairwise orthogonal and summable, then $x = x_1+\dots+x_n \in$ I(G) and

$$G = \mathrm{Kerx}\,\lambda\,\mathrm{Imx},$$

$$\mathrm{Imx} = \mathrm{Imx}_1 \times \dots \times \mathrm{Imx}_n \text{ (the direct product)},$$

$$\mathrm{Kerx} = \bigcap_{i=1}^{n} \mathrm{Kerx}_i.$$

In the lemmas 4–8 below it is supposed that G and G* are suitable groups for which the semigroups of endomorphisms are isomorphic:

$$\mathrm{EndG} \cong \mathrm{EndG^*}. \qquad (1)$$

The image of an element z of End G by isomorphism (1) is denoted everywhere as z*.

LEMMA 4 ([1], lemma 2). If x $\in$ I(G), then the semigroups End(Imx) and End(Imx*) are isomorphic.

LEMMA 5 ([3], corollary 1.12). If x$\in$I(G) and G = = Kerx$\times$Imx, then G* = Kerx*$\times$Imx*.

From lemma 2 it follows:

LEMMA 6. For x,y$\in$I(G) the isomorphism (1) maps the set Hom(Imx, Imy) one-to-one on the set Hom(Imx*, Imy*).

LEMMA 7 ([4], lemma 5). If $x_1,\dots,x_n \in I(G)$ and the idempotents $x_1,\dots,x_n$ are pairwise orthogonal and summable, then their images $x_1^*,\dots,x_n^*$ are also pairwise orthogonal and summable and

$$(x_1+\dots+x_n)^* = x_1^*+\dots+x_n^*.$$

LEMMA 8 ([5], corollary 1). Suppose that $x,y \in I(G)$, x and y are orthogonal and summable, $u \in Hom(Imx, Imy)$ and $Imu \subset Z(Imy)$, then $Imu^* \subset Z(Imy^*)$.

3. Proof of Theorem

Assume further that G is a reduced Abelian group with torsion subgroup T(G). Let G* be another group such that semigroups End G and End G* are isomorphic:

$$\text{End } G \simeq \text{End } G^*. \qquad (2)$$

Denote by π a set of all prime divisors of orders of elements of T(G). If set π is empty, then the statements of theorem are trivially true. Therefore it is further assumed that π is non-empty.

It is sufficient to show that for any prime $p \in \pi$ the following statements hold:

(A) the center Z(G*) of G* contains all p-elements of G*;

(B) the subgroup G_p^* of all p-elements of G^* and the subgroup G_p of all p-elements of G are isomorphic.

Suppose now that $p \in \pi$, then the following three lemmas are true.

LEMMA 9. If a torsion subgroup B of a p-basic subgroup of G is bounded, then Z(G*) contains all p-elements of G* and groups G_p and G_p^* are isomorphic.

PROOF. Let B be a torsion subgroup of a p-basic subgroup of G and assume that B is bounded. Then G splits into a direct sum:

$$G = B \oplus G_1. \qquad (3)$$

There are no non-trivial p-elements in group G_1. Therefore,

$$Hom(B, G_1) = \{0\}. \qquad (4)$$

By lemmas 4 and 5 it implies now that G* splits as

$$G^* = B^* \times G_1^* \qquad (5)$$

where

$$\mathrm{End}\, B \simeq \mathrm{End}\, B^*$$

(take x as a projection $G \longrightarrow B$). Since every bounded Abelian group is generated by its semigroup of all endomorphisms in the class of all groups (see [6], theorem), groups B and B^* are isomorphic.

From lemma 6 and equation (4) it follows that group G_1^* has no non-trivial p-elements. Hence, $G_p = B \simeq B^* = G_p^*$. It follows now in view of (5) that $G_p^* = B^* \subset Z(G^*)$. This completes the proof of the lemma.

Assume further that the torsion subgroup of the basic p-subgroup of group G is non-bounded. Then there exist elements $a_1, a_2, a_3, \ldots$ of G such that

$$G = \langle a_1 \rangle \oplus \ \ldots \ \oplus \langle a_k \rangle \oplus G_k, \qquad (6)$$

$$G_k = \langle a_{k+1} \rangle \oplus G_{k+1}, \qquad (7)$$

$$\langle a_k \rangle \simeq C_{p^{n_k}} \qquad (8)$$

for every integer k, and

$$1 \leq n_1 < n_2 < n_3 < \ldots \ . \qquad (9)$$

We denote further by ε_k a projection of group G onto their subgroup $\langle a_k \rangle$.

From lemmas 7 and 3-5 it follows that group G^* splits as following:

$$G^* = \mathrm{Im}\, \varepsilon_1^* \times \ \ldots \ \times \mathrm{Im}\, \varepsilon_k^* \times G_k^* \qquad (10)$$

where

$$G_k^* = \bigcap_{i=1}^{k} \mathrm{Ker}\, \varepsilon_i^*,$$

$$G_k^* = \mathrm{Im}\, \varepsilon_{k+1}^* \times G_{k+1}^*, \qquad (11)$$

$$\mathrm{End}(\mathrm{Im}\, \varepsilon_k^*) \simeq \mathrm{End}\, \langle a_k \rangle \qquad (12)$$

and ε_k^* denotes the image of ε_k by isomorphism (2). From isomorphism (12) the isomorphism $\mathrm{Im}\, \varepsilon_k^* \simeq \langle a_k \rangle$ follows (see [6], theorem).

Let c_k be a generator of group $\mathrm{Im}\, \varepsilon_k^*$. Then in view of (10)-(12) we have

$$G^* = \langle c_1 \rangle \times \dots \times \langle c_k \rangle \times G_k^*, \quad (13)$$

$$G_k^* = \langle c_{k+1} \rangle \times G_{k+1}^*, \quad (14)$$

$$\langle c_k \rangle \simeq C_{p^{n_k}}. \quad (15)$$

LEMMA 10. If g is a p-element of G^*, then $g \in Z(G^*)$, i.e. $G_p^* \subset Z(G^*)$.

PROOF. Let g be a p-element of the group G^* and p^n the order of g. By (9) there exists an integer t such that $n_t \geqslant n$. Then element g can be presented as

$$g = bc, \ b \in \operatorname{Im} \varepsilon_t^* = \langle c_t \rangle, \ c \in \operatorname{Ker} \varepsilon_t^*.$$

Let z^* be an endomorphism of group G^* such that

$$c_t z^* = c; \ hz^* = 1 \quad \text{where} \ h \in \operatorname{Ker} \varepsilon_t^*.$$

Since $n_t \geqslant n$ and conditions (10)-(15) are true, the endomorphism z^* is defined correctly. By lemma 8 (where $x^* = \varepsilon_t^*$, y^* is the projection of G^* onto Ker ε_t^* and $u^* = z^*$), element c is contained in $Z(\operatorname{Ker} \varepsilon_t^*)$. From conditions (10)-(14) it follows now that $b, c \in Z(G^*)$ and $g = bc \in Z(G^*)$. This completes the proof of the lemma.

LEMMA 11. Groups G_p and G_p^* are isomorphic.

PROOF. Since $a_k \in G_p$ and $c_k \in G_p^*$, there exists a subgroup A_k of G_p and a subgroup B_k of G_p^* such that

$$G_p = \langle a_1 \rangle \oplus \dots \oplus \langle a_k \rangle \oplus A_k,$$

$$G_p^* = \langle c_1 \rangle \oplus \dots \oplus \langle c_k \rangle \oplus B_k,$$

$$A_k = \langle a_{k+1} \rangle \oplus A_{k+1},$$

$$B_k = \langle b_{k+1} \rangle \oplus B_{k+1}.$$

Therefore the construction of the isomorphism between groups G_p and G_p^* coincides with the construction of the isomorphism between groups A and B in lemma 12 of [1]. The lemma is proved.

This completes also the proof of our theorem.

References

1. Пуусемп П. Об определяемости периодической абелевой группы своей полугруппой эндоморфизмов в классе всех периодических абелевых групп // Изв. АН Эст. ССР. Физака. Математика. 1980. 29. № 3. С. 246-253.
2. Курош А.Г. Теория групп. М.: Наука, 1967.
3. Пуусемп П. Идемпотенты полугрупп эндоморфизмов групп // Уч. зап. Тартуск. ун-та. 1975. № 366. С. 76-104.
4. Пуусемп П. Об одной теореме Мэя // Изв. АН Эст. ССР. Физика. Математика. 1989. 38. № 2. С. 139-145.
5. Пуусемп П. О суммируемости эндоморфизмов групп // Тезисы докладов конференции "Методы алгебры и анализа". Тарту. 1988. С. 24-27.
6. Пуусемп П. Об определяемости периодической абелевой группы своей полугруппой эндоморфизмов // Изв. АН Эст. ССР. Физика. Математика. 1980. 29. № 3. С. 241-245.

P. Puusemp

REDUCED ABELIAN GROUPS AND THEIR SEMIGROUPS OF ENDOMORPHISMS

Abstract

For a suitable group G let $\pi(G)$ be a set of all prime divisors of orders of elements of finite order of group G. The following theorem is proved.

THEOREM. Let G be a reduced Abelian group and T(G) the torsion subgroup of group G. Let G^* be a group such that the semigroups of all endomorphisms of groups G and G^* are isomorphic. Then all the $\pi(G)$-elements of G^* belong to the center of G^* and form a subgroup isomorphic to T(G).

From this theorem it directly follows:

COROLLARY. Suppose that G and G^* are two reduced Abelian groups such that the semigroups of all endomorphisms of G and G^* are isomorphic. Then the torsion subgroups T(G) and $T(G^*)$ of G and G^* are also isomorphic.

P. Puusemp

REDUTSEERITUD ABELI RÜHMAD JA NENDE ENDOMORFISMIPOOLRÜHMAD

Kokkuvõte

Tähistame sümboliga $\pi(G)$ rühma G lõplikku järku elementide järkude algarvuliste tegurite hulka. Kommutatiivse rühma G korral tähistame sümboliga T(G) rühma G perioodilist osa, s.t. rühma G lõplikku järku elementidest koosnevat alamrühma. Artiklis tõestatakse:

TEOREEM. Kui redutseeritud Abeli rühma G endomorfismipoolrühm on isomorfne mingi rühma G^* endomorfismipoolrühmaga, siis rühma G^* kõik $\pi(G)$-elemendid sisalduvad rühma G^* tsentris ja moodustavad rühmaga T(G) isomorfse alamrühma.

JÄRELDUS. Kui kahe redutseeritud Abeli rühma endomorfismipoolrühmad on isomorfsed, siis nende rühmade perioodilised osad on samuti isomorfsed.

No 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 519.8+519.6

E. Übi

THE SECOND METHOD FOR SOLVING THE PROBLEM OF $Ax = b$, $x \geq 0$ BY ORTHOGONAL TRANSFORMATIONS

1. Introduction

Let A be a mxn-matrix and b a m-vector. We have to find n-vector x which satisfies conditions

$$Ax = b$$
$$x \geq 0. \qquad (1)$$

Let us compose an auxiliary problem of the least-squares

$$\varphi(x) = \frac{1}{2} \| Ax - b \|^2 \longrightarrow \min_{x \geq 0}, \qquad (2)$$

where $\| \cdot \|$ is the Euclidean norm. In a non-singular case of the problem we have to determine m linear independent columns of matrix A, which corresponds to the active variables x_j. At the beginning all the variables are passive, $x^0 = 0$. Similarly [1] on every step of method VRM3 either according to the formula (3), a new active variable is determined, or one of the active variables (which becomes negative) is substituted by a new one.

Matrix A is not conversed in the process of computations. So we use an auxiliary matrix R to store all the current information on the transformed active columns. The last (m+1)-th column of R is used for memorizing the right-hand b which was conversed [2](chapter 24).

At activation of a new variable $x_{j_{k+1}}$ the following problem

$$\frac{\partial\varphi(x^k)}{\partial x_j} : \|a_j\| \longrightarrow \min_{j=1,\dots,n} \qquad (3)$$

is solved, where x^k is a current point determined by the least-squares method from the overdetermined system with k variables and m equations. $x^o = 0$, $k = 0,1,\ldots,$ [1]. To find the next point x^{k+1} we write column $a_{j_{k+1}}$ in (k+1)-th column of matrix R. In the upper triangular part of the first columns of R there exist active columns which are posted on the previous steps and conversed into the triangular form by Householder reflection transformation. Instead of storing the zero elements in the subdiagonal part of R, we memorize information of the fulfilled Householder reflections. If the number of the active variables is equal to k, then it is necessary to memorize m - k + 1 elements of a normal u(k) hyperplanes of reflection mapping. Since the number of elements below the main diagonal of R is equal to m-k, then on each step it is necessary to post in the additional vector W of dimension m the k-th normal element, [2] (chapter 10). To find the next point x^{k+1}, we have to post column $a_{j_{k+1}}$ in the (k+1)-th column of the matrix R. For this particular column all the k previous Householder transformations are fulfilled and diagonal elements are posted in array W. Further, based on the (k+1)--th column of R a Householder transformation is made which nullifies the elements in the (k+1)-th column beginning with the (k+2)-th element. The elements themselves in the column, excluding (k+1)-th, are not changed by the last conversion. They are used at the subsequent steps as normals' components of the hyperplane reflection transformation. Finally a new point x^{k+1} is determined by the triangular system. Changing a variable in the case $x_j^k < 0$, on the following step x_j is eliminated and a new variable is activated. Let x_j be l-th in the list of the active variables. Then for its elimination we remove from R all converted active columns beginning with l-th and the right-hand part in the column (m+1)-th as well. They are substituted by the corresponding non-transformed columns A and b. When the operation is over matrix R is given again a triangular form. To achieve this, the substituted columns R and b are subjected to the l-1 previous orthogonal transformations which are determined by l-1 first columns of R and an array W. Let us denote submatrix R, composed of these transformed columns, by S. Then by l-th, (l+1)-th, ..., (k+1)-th of the column R are performed specially for S and for the right-hand part the Householder reflection transformation deletes elements of the first column S beginning with (l+1)-th, the second with (l+2)-th, etc. As it was stated above, instead of zero-elements under the main diagonal R normals' components of transformations of these reflections are memorized. In conclusion, from the triangular system we determine values of the active variables and by them, accordingly, a new active variable and the next point x^{k+1}.

The proposed method corresponds to the first version of the second method [2], chapter 24. For the second version of this method a subprogram VRM4 was composed. However, it proved less efficient than VRM3. In the subprogram VRM4 the other scheme for computing matrix S in the case $x_j^k < 0$: columns R with numbers l+1,...,k are subjected to the orthogonal conversions, performed on k-th, (k-1)-th,..., l-th steps, i.e. the situation which preceded the activation of x_j on l-th step is completely reset. By matrix S found in this way the further computations are made as in the first version.

2. Description of Algorithm

Algorithm VRM3 can be applied also for the problems with a sparse matrix A, where in addition to arrays A and b, an auxiliary mx(m+1)-matrix R, two m-vectors v,w and three n-vectors F,G,x are needed.

If in the first algorithm VRM1 the converted system (1) was used to determine an input variable by- formula (3), then in algorithm VRM3 the initial system was used. In the additional array v we memorize the right hand side of the system (1), evaluated according to the formula $b - \sum_{j \in IJ} a_{ij} x_j$, where summation is performed over all active variables. In contrast to the first method, there is no possibility to make recurrence computations of derivative functions $\varphi(x)$ on every step.

Let us describe algorithm VRM3 in detail to solve the problem (1).

The algorithm VRM3 $(A,b,R,IJ,v,w,x,F,G,m,n,\varepsilon_1,\varepsilon_2)$.

1. Evaluate n-vector G with coordinates

$$G_j = \| a_j \| = \sqrt{\sum_{i=1}^{m} a_{ij}^2},\ j=1,\dots,n. \qquad (4)$$

2. Write into (m+1)-th column of the matrix R the right hand side b,

$$R_{i\,m+1} = b_i,\ i=1,\dots,m.$$

3. Initially the number of active variables $k:=0$.

4. Find m-vector v with coordinates

$$v_i = b_i - \sum_{j\in IJ} a_{ij} x_j,\ i=1,\dots,m, \qquad (5)$$

where summation is performed over all k active variables. (On the first step the set of indices of the active variables IJ is empty.)

5. Compute inner products

$$F_j = \sum_{i=1}^{m} a_{ij} v_i,\ j=1,\dots,n. \qquad (6)$$

6. Determine the following active variable $x_{j_{k+1}}$ by the solution of the problem

$$\max_{j=1,\dots,n} \frac{F_j}{G_j} = \frac{F_{j_{k+1}}}{G_{j_{k+1}}} = RE. \qquad (7)$$

Write index j_{k+1} into the array IJ of the indices of active variables.

7. If $RE \geq \varepsilon_1$, go to step 10.

8. Check inequalities $|R_{im+1}| \leq \varepsilon_2$ for $i=k+1,\dots,m$. If all of them are fulfilled, go to step 28 or else go to step 9.

9. The system (1) has no solution. Stop.

10. Increase number of active variables, $k:=k+1$.

11. Write a new active column.

$$R_{ik} := A_{ij_k},\ i=1,\dots,m, w(k):=R(k,k).$$

12. At $k \geqslant 2$ to perform k-th column R the previous k-1 Householder transformations.

13. If the inequality $k < m$ holds, go to step 14, else go to step 16.

14. Compute a new pivoting element R_{kk}.

15. Fulfil Householder transformation with a vector $v = R_k$ to the right hand side R_{m+1}.

16. Solve the triangular system

$$R(k)\, x = R_{m+1}$$

of order k to determine the active variables.

17. Check whether the active variables are non-negative, $x_j \geqslant 0$, $j \in IJ$. If all inequalities hold, go to step 26.

18. If $x_j < 0$ and $IJ(l) = j$, delete the active index j and initially $x_j := 0$.

19. Substitute columns $R_{l+1}, \ldots, R_{k-1}$ with the corresponding columns A and $R_{m+1} := b$.

20. Do columns $R_{l+1}, \ldots, R_{k-1}$ and R_{m+1} the previous l-1 Householder transformations determined by the subdiagonal part of the first l-1 columns R and array W.

21. Do Householder transformations with pivoting column $v = R_j$ for columns $R_{j+1}, \ldots, R_{k-1}$ and R_{m+1}, $j = l, l+1, \ldots, k-1$, memorizing in the subdiagonal part R and in array W the normals of these reflections.

22. After elimination of the l-th element initially

$$IJ(t) := IJ(t+1),\ t = l, \ldots, k-1.$$

23. Decrease the number of the active variables, $k := k-1$.

24. Solve the triangular system $R(k)x = R_{m+1}$ of the order k to determine the active variables.

25. Go to step 4.

26. If the inequality $k < m$ holds, go to step 27, else go to step 28.

27. Check inequalities $|R_{im+1}| \leqslant \varepsilon_2$ for $i = k+1, \ldots, m$. If all of them hold, go to step 28, else go to step 4.

28. The problem is solved.

Remarks. 1. If at a certain step some variables x_j obtain negative values, then to simplify the calculus the variable which has activated last is removed. The other negative variables may become positive after removal of $x_j < 0$.

2. In the first algorithm VRM1 the derivative function is calculated at every step by simple recurrence formulas based on the converted system (1). In algorithm VRM3 the passive columns A are not transformed, therefore to compute derivative $\varphi'(x)$ it is necessary to find at every step a new right hand part $\tilde{b} = b - \sum_{j \in IJ} a_{ij} x_j$ of the initial system and its scalar products on the passive columns. However, solving many problems with a sparse matrix A and with identical position non-zero elements it is possible to decrease considerably the size of calculus for finding $\varphi'(x)$.

3. In this algorithm the column norms G_j are not recomputed, since for computations of $\varphi'(x)$ the initial system is used. When determining column A which forms a minimal angle with a right hand side b they are considered as m-vectors (as distinguished from VRM1, where dimension m-k of columns was decreasing).

3. Numerical Results

Example 1.

$$\begin{aligned} -x_2 + x_3 &= 0 \\ x_1 + x_2 + x_3 - x_4 &= 1.5 \\ x_2 &= 1 \\ x &\geq 0. \end{aligned} \qquad (8)$$

The solution of the problem by algorithm VRM1 is given in [1]. The variables are activated in the following order: x_1, x_2, x_3. On the fourth step $x_1 < 0$ is deleted and x_4 activated. The solution of the problem $x_2 = x_3 = 1$, $x_4 = 0.5$. In the table 1 contents of the arrays R, W, F, G, x are presented.

Step		x_1	x_2	x_3	x_4	b	w	x
	F	1.50	2.50	1.50	-1.50			
	G	1	1.73	1.14	1			
		-1				-1.50	0	1.50
1	R	1				0		
		0				1		
	F	0	1	0	0			

Step		x_1	x_2	x_3	x_4	b	w	x
		-1	-1			-1.50	0	1
	R	1	-1.41			-0.71	1	0.5
2		0	1			0.71		
	F	0	0	0.50	0			
		-1	-1	-1				-0.50
	R	1	-1.41	0.71				1
3		0	1	0.71				1
	F	0	0	0	0			
			1.73	0	-0.58	1.44	-1	1
4	R		1	-1.41	0.71	-1.06	1.36	1
			1	0.37	0.41	0.20	0	0.50
	F		0	0	0.08			

Table 1

Remarks.

4. At the first step x_1 is activated, components of the normal of the first reflection are $u_1(1)=w_1=0$, $u_2(1) = R_{21} = 1$, $u_3(1) = R_{31} = 0$, and $x_1 = 1.50$.

5. At the third step the non-negativity condition is violated, $x_1 = -0.5 < 0$. Therefore the first column is eliminated, the second and the third columns of R and the right hand side are substituted with the corresponding columns A and b. Householder transformations which conversed the second and third columns into a diagonal form are used here. Finally, at the fourth step x_4 is activated.

6. The time required for solving examples by algorithms VRM1 and VRM3 which are given above in [1], where $mn \leq 45\,000$, is almost the same. To solve problems with the sparse matrix A if mn exceeds capacity of a core memory only algorithm VRM3 is used.

7. The experience with high dimension states shows that though values F_j are changing with an increase of the number of the step k, the order of relations F_j/G_j is not hardly affected. Therefore, provided the number of equations is $m \geq 100$, then at step 6 of algorithm VRM3, it is possible to determine simultaneously several variables which correspond to the greater relations of F_j/G_j and make them active. That allows the amount of computations to be reduced. In this paper we do not go into details.

References

1. Übi E. Application of the Least Squares Method in Mathematical Programming. News No 4. Academy of Sciences of the Est. SSR. Physics. Mathematics, 1989.

2. Lawson C., Hanson R. Solving Least Squares Problems, 1974. /Prentice-Hall. Inc. Englewood Cliffs. New-Jersey/. (Russian translation of the book is made in 1989).

E. Übi

THE SECOND METHOD FOR SOLVING THE PROBLEM OF $Ax=b$, $x \geq 0$ BY ORTHOGONAL TRANSFORMATIONS

Abstract

According to the first scheme, suggested for the application of the least-squares method for an underdetermined system of linear equations all the coefficients of the system were subjected to orthogonal transformations on every step. In the present paper we propose a different approach, which can be used also for a compact set of a sparse matrix A. In the computing process A is not conversed. We use an auxiliary matrix R to store triangular system for determining current values of variables x_j. Normals of hyperplanes reflection are stored in the subdiagonal part of R. These normals serve for memorizing as a product, the orthogonal matrix which corresponds to the fulfilled transformations.

E. Übi

TEINE MEETOD ÜLESANDE $Ax = b$, $x \geq 0$ LAHENDAMISEKS ORTOGONAALTEISENDUSTE ABIL

Kokkuvõte

Kitsenduste maatriksit A ei teisendata, vaid kasutatakse abimaatriksit R, mis sisaldab arvutusteks vajalikku informatiooni. Esitatav meetod on rakendatav hõreda maatriksi A korral.

№ 723

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED

ТРУДЫ ТАЛЛИННСКОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

ОБРАБОТКА ДАННЫХ. ПОСТРОЕНИЕ ТРАНСЛЯТОРОВ. ВОПРОСЫ ПРОГРАММИРОВАНИЯ

Труды по экономике XXУ

УДК 681.06

Экспертная система как часть информационной системы. Выханду Л., Куузик Р. - Труды Таллиннского технического университета. 1991. № 723, с. 3-7.

В данной статье описывается применение теории монотонных систем при создании экспертных систем. Приводится процесс генерации фактов и правил из реальной базы данных. При этом остается в силе одно из основных требований инфосистем - описание исследуемой совокупности данных наименьшим количеством признаков.

Таблиц - I, библ. наименований - 3.

УДК 681.06

Наилучшая ранжировка в турнирах. Выханду Л. - Труды Таллиннского технического университета. 1991. № 723, с. 8-14.

В статье предлагается быстрый метод получения стабильной ранжировки в турнирах и его алгоритм.

Таблиц - 9, библ. наименований - 4.

УДК 681.3.06

Человеческие факторы в разработке экспертных систем. Стучлик Ф., Тепанди Я. - Труды Таллиннского технического университета. 1991. № 723, с. 15-25.

Экспертные системы (ЭС) оправдывают себя только в плохо структурированных, неформализованных областях. Разработка таких систем требует усиленного взаимодействия разработчиков и пользователей, следовательно, роль человеческого фактора здесь особенно важна. Мы характеризуем интересы пользователей ЭС, анализируем задачи ключевых ролей в валидации ЭС, изучаем оптимальную структуру человеческих понятий для баз знаний ЭС.

Библ. наименований - 13.

УДК 681.3.06

Генерирование и визуализация тестов в инструментальной экспертной системе ХЕЛИ. Пармаксон П. - Труды Таллиннского технического университета. 1991. № 723, с. 26-40.

В статье анализируются методы и средства генерирования и визуализация тестов в инструментальной экспертной системе ХЕЛИ. Приводится краткое описание алгоритма генерирования эффективных тестов. Рассматриваются связи между задачами генерирования тестов и проверки статистической коррекции базы знаний.

Рис. - 1, библ. наименований - 8.

УДК 681.06

Обработка структурных данных с помощью систем построения трансляторов. Луйге Т. - Труды Таллиннского технического университета. 1991. № 723, с. 41-52.

В данной статье рассматриваются основы ввода сложных текстовых форм как структурных объектов. Более подробно описывается система ввода и автоматического распознавания элементов библиографических данных. Ее теоретическая часть основывается на формальных грамматиках и языках. Базой для программирования является система построения трансляторов ELMA.

Рис. - 2, библ. наименований - 6.

УДК 681.3.016

Планирование организации в понятиях проектирования информационной системы. Лумберг Т. - Труды Таллиннского технического университета. 1991. № 723, с. 53-63.

В статье рассматриваются проблемы планирования организации с помощью терминов проектирования информационной системы. Характеризуется набор концептов начальных стратегических решений.

Библ. наименований - 10.

УДК 681.06

Контора как сложная система. Ельмик Л. - Труды Таллиннского технического университета. 1991. № 723, с. 64-73.

В данной работе рассматриваются проблемы моделирования конторской информационной системы. Выводятся определение конторы и модели ее отношений в процессе управления.

Рис.- 7, библ. наименований - 14.

УДК 519.4

Редуцированные абелевы группы и их подгруппы эндоморфизмов. Пуусемп П. - Труды Таллиннского технического университета. 1991. № 723, с. 74-80.

Обозначим через $\pi(G)$ совокупность всех простых чисел, являющихся делителями порядков элементов конечного порядка группы G. В статье доказывается

ТЕОРЕМА. Если G - редуцированная абелева группа с периодической частью $T(G)$ и полугруппа всех эндоморфизмов группы G изоморфна полугруппе всех эндоморфизмов некоторой другой группы G^*, то все $\pi(G)$-элементы группы G^* образуют подгруппу, находящуюся в центре группы и изоморфную группе $T(G)$.

Из этой теоремы сразу следует

СЛЕДСТВИЕ. Если полугруппы всех эндоморфизмов двух редуцированных абелевых групп изоморфны, то периодические части этих групп изоморфны.

Библ. наименований - 6.

УДК 519.8+519.6

Второй метод решения задачи $Ax=b,\ x \geq 0$ с помощью ортогональных преобразований. Юби Э. - Труды Таллиннского технического университета. 1991. № 723, с. 81-88.

Для этой недоопределенной системы линейных уравнений в первой [1] предложенной схеме применения метода наименьших квадратов (НК) на каждом шаге подвергались ортогональным преобразованиям все коэффициенты системы. В данной работе предлагается другой способ вычислений, применимый и в случае компактной записи разреженной матрицы A. В ходе вычисления A не преобразуется. Используется вспомогательная матрица R, где запоминается треугольная система для определения текущих значений переменных x_j. В поддиагональной части R хранятся нормали гиперплоскостей отражения. С помощью этих нормалей запоминается в виде произведения ортогональная матрица, соответствующая выполненным преобразованиям.

Библ. наименований - 2.