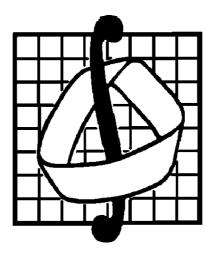
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В. ЛОМОНОСОВА



Механико-математический факультет

МАТЕРИАЛЫ IX Международной конференции "Интеллектуальные системы и компьютерные науки"

(23-27 октября 2006 г.)

 $\underset{\text{часть }1}{\text{TOM}} 1$

Издание осуществлено при поддержке Российского фонда фундаментальных исследований по проекту № 06-01-10-114

Материалы IX Международной конференции "Интеллектуальные системы и компьютерные науки" (23-27 октября 2006 г.), том 1, часть 1. - М.: Изд-во механикоматематического факультета МГУ, 2006.

Сборник содержит работы участников IX Международной конференции "Интеллектуальные системы и компьютерные науки", проходившей на механикоматематическом факультете МГУ им. М. В. Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114). Сборник адресован научным сотрудникам, преподавателям, аспирантам и студентам, работающих и интересующихся тематикой математических проблем теории интеллектуальных систем и их приложений.

Научное издание

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИ-СТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ" (23-27 октября 2006 г.) Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В.

В составлении и редактировании сборника принимали участие: Строгалов А. С., Носов В. А., Уварова Т. Д., Холоденко А. В., Галатенко А. А.

Ответственный за выпуск Строгалов А. С.

© Механико-математический факультет МГУ, 2006

ПРЕДИСЛОВИЕ

IX Международная конференция «Интеллектуальные системы и компьюерные науки», материалы которой публикуются в настоящих сборниках (т.1 и т.2) проходила на механико-математическом факультете МГУ им. М.В.Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114).

Оргкомитетом конференции до начала ее работы были разосланы информационные письма в ведущие научные центры и университеты России и стран СНГ, в ряд зарубежных научных центров и университетов, отобраны наиболее интересные доклады и сообщения для заслушивания их на пленарных и секционных заседаниях.

В работе конференции принимало участие около 300 участников (ведущие ученые, преподаватели, научные сотрудники, аспиранты и студенты) из Азербайджана, Алжира, Армении, Белоруссии, Великобритании, Германии, Грузии, Иордании, Казахстана, Канады, Киргизии, КНР, Пакистана, России, Сербии, Сирии, Словении, США, Узбекистана, Украины, Франции, Черногории, Японии.

Работа конференции проходила по следующим направлениям:

- распознавание образов;
- информационные процессы и технологии;
- интеллектуальное программное обеспечение;
- базы данных и знаний;
- принципы принятия решений;
- экспертные системы и решатели задач;
- моделирование систем и процессов в технике, естественных и гуманитарных науках;
- дискретная оптимизация;
- теория дискретных функций;
- автоматы и роботы с элементами искусственного интеллекта;
- компьютеры и нейрокомпьютеры;
- нечёткая математика и её приложения;
- компьютерная алгебра;
- теория кодирования и криптография;
- параллельные вычисления

Содержание пленарных и секционных докладов отражены в настоящих материалах конференции. Тексты публикуются в авторской редакции (исправлены замеченные опечатки).

Belief Propagation Algorithm for Traffic Prediction

Cyril Furtlehner ^{1 2} Arnaud de La Fortelle^{1 2} Jean-Marc Lasgouttes¹

The aim of this paper is to give an overview of a new method for traffic prediction. This method is particularly suitable in settings where the only information available is floating car data. We propose a system, based on the Ising model of statistical physics, which both reconstructs and predicts the traffic in real time using a message-passing algorithm. The realisation and the future developments are briefly presented.

Keywords: belief propagation algorithm, Bethe approximation, traffic prediction, intelligent traffic systems, floating car data.

Introduction

With an estimated 1% GDP cost in the European Union (i.e. more than hundred billions euros), congestion is not only a time waste for drivers and an environmental challenge, but also an economic issue. This is why the European commission financed the REACT project, where new traffic prediction models have been developed. These predictions are to be used to inform the public and possibly to regulate the traffic.

We want to present some constraints to take into account for integrating our algorithm in a real system. First consideration, the information available for traffic reconstruction and prediction is very heterogeneous. Sensors can be magnetic loops, video cameras or floating car data (FCD), the data retrieved and send by a car to a server. These data do not belong usually to a unique "information operator" and thus formats, reliability and availability generate lots of noise for analysis.

Second consideration, today, some urban and inter-urban areas have traffic management and advice systems that collect data from stationary sensors, analyse them, and post notices about road conditions ahead and recommended speed limits on display signs located at various points along specific routes. However, these systems are not available everywhere and they are virtually non-existent on rural areas. With rural road crashes accounting for more than 60% of all road fatalities in OECD (Organisation for Economic Cooperation and Development) countries, the need for a system that can cover these roads is compelling if a significant reduction in traffic deaths is to be achievable.

The REACT project combines a traditional traffic prediction approach on equipped motorways with an innovative approach on non-equipped roads. The idea is to obtain floating car data from a fleet of probe vehicles and reconstruct the traffic conditions from this partial information. To understand why it is not possible to fuse these two parts, we have to go a bit more into prediction algorithms details.

Two types of approaches are usually distinguished, namely data driven (application of statistical models to a large amount of data, for example regression analysis) and model based (simulation or mathematical models explaining the traffic patterns). As we stated before, the choice is largely led by the availability of data. In our case, since few data is available on non-equipped roads (only the equipped vehicles driving along the observed roads), the model driven approach is the only feasible one. For more information about traffic prediction methods, we refer the reader to [1,8,9].

Most current traffic models are deterministic, described either at a macroscopic level by a set of differential equations linking variables such as flow and density, or by Newton's law at a microscopic level where each individual car is considered. Intermediate descriptions are essentially kinetic models, like for example cellular automata [6], which are very well adapted to freeway traffic modelling and adapted to some extent to urban traffic modelling [2]. Traffic flow models are quite adapted and efficient on motorways where fluid approximation of the traffic is reasonable; they tend to fail for cities or rural roads. The reason is that the velocity flow field is subject to much greater fluctuations induced by the nature of the network (presence of intersections and short distance between two intersections) than by the traffic itself. These fluctuations are both spatial and temporal (a red or green traffic light at a cross-road, a road-work, etc). There is no local stationary regime for the velocity, the dynamics are dominated by the fluctuations. This requires a stochastic model, which is what we describe in the next section.

Traffic description and statistical physics

¹INRIA Rocquencourt – Domaine de Voluceau B.P. 105 – 78 153 Le Chesnay Cedex (France)

 $^{^2}$ cole des Mines de Paris – CAOR research centre – 60, boulevard Saint-Michel – 75272 Paris cedex 06

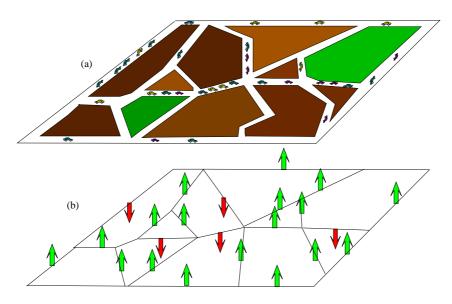


Figure 1: Traffic network (a) and Ising model (b) on a random graph

We propose in this paper a hybrid approach, by taking full advantage of the statistical nature of the information, in combination with a stochastic modelling of traffic patterns. In order to reconstruct the traffic and make predictions, we propose a model—the Bethe approximation (BA)—to encode the statistical fluctuations and stochastic evolution of the traffic and an algorithm—the belief propagation (BP) algorithm—to decode the information. Those concepts are familiar to the computer science and statistical physics communities since it was shown [10] that the output of BP is in general the Bethe approximation.

The graph onto which we apply the belief propagation procedure is made of space-time nodes that encode both a location (road link) and a time (discretized on a few minutes scale). More precisely, the set of nodes is $\mathcal{N} = \mathcal{L} \otimes \mathbb{Z}^+$, where \mathcal{L} corresponds to the links of the network and \mathbb{Z}^+ to the time discretization. To each point $\alpha = (\ell, t) \in \mathcal{N}$, we attach an information $\tau_{\alpha} \in \{0, 1\}$ indicating the state of the traffic (1 if congested, 0 otherwise). Each cell is correlated to its neighbour (in time and space) and the evaluation of this local correlation determines the model. In other words, we assume that the joint probability distribution over the set is of the form

$$p(\{\tau_{\alpha}, \alpha \in \mathcal{N}\})$$

$$= \prod_{\alpha \in \mathcal{N}} \psi_{\alpha}(\tau_{\alpha}) \left[\prod_{\beta \in V(\alpha)} \psi_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}) \right]$$

$$(1)$$

where $V(\alpha) \subset \mathcal{N}$ is the set of neighbours of α and the local correlations are encoded in the functions ψ . \mathcal{N} together with V describe the space-time graph \mathcal{G} .

The model described by (1) is actually equivalent to an Ising model [3] on \mathcal{G} , with arbitrary coupling between adjacent spins, the up or down orientation of each spin indicating the status of the corresponding link (Figure 1).

The homogeneous Ising model (uniform coupling constants) is a well-studied model of ferro (positive coupling) or anti-ferro (negative coupling) material in statistical physics. It displays a phase transition phenomena with respect to the value of the coupling. At weak coupling, only one disordered state occurs, where spins are randomly distributed around a mean-zero value. Conversely, when the coupling is strong, there are two equally probable states that correspond to the onset of a macroscopic magnetisation either in the up or down direction: each spin has a larger probability to be oriented in the privileged direction than in the opposite one.

From the point of view of a traffic network, this means that such a model is able to describe three possible traffic regimes: fluid (most of the spins up), congested (most of the spins down) and dense (roughly half of the links are congested). For real situations, we expect other types of congestion patterns, and we seek to associate them to the possible states of an inhomogeneous Ising model (i.e. with general coupling parameters), which has also been well studied in statistical physics, in the context of spin-glasses [5]. Indeed in some cases, when the system is frustrated, because some of the couplings are negative, leading to a certain number of contradictions, a proliferation of meta-stable states occurs, which eventually scales exponentially with the size of the system.

In practice what we retain from an Ising description is the possibility to encode a certain number of traffic patterns in a statistical physics model. This property is actually shared also by the BA and this is a the reason for us to directly encode the traffic patterns in a BA rather than a inhomogeneous Ising model, based on historical data, and to avoid therefore an intermediate approximation step. BA simply provide us with a set of marginals probabilities. But this set which is the result of an iterative procedure, is not necessarily unique (see for example [4]) and the proliferation of possible solutions depends on the frustration induced by the historical correlations used to define the ψ 's of (1). The setting of our model consists therefore into an optimisation procedure of the matching between the set of historical values obtained from probe vehicles and the set of marginal probabilities of the BA.

The data collected from the probe vehicles is used in two different ways. The most evident one is that the data of the current day directly influences the prediction. In parallel, this data is collected over long periods (weeks or months) in order to estimate the model (1). Typical historical data that is accumulated is

- $\hat{p}_{\alpha}(\tau_{\alpha})$: the probability that node α is congested $(\tau_{\alpha} = 1)$ or not $(\tau_{\alpha} = 0)$;
- $\hat{p}_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta})$: the probability that a probe vehicle goes from α with state τ_{α} to β with state τ_{β} .

The reconstruction and prediction algorithm

The belief propagation algorithm

BP is a message passing procedure, which output is a set of estimated marginal probabilities, the beliefs $b_{\alpha\beta}[7]$. The idea of the BP algorithm is to factor the marginal probability at a given site in a product of contributions coming from neighbouring sites, which are the messages. The messages sent by a node α to $\beta \in V(\alpha)$ depends on the messages it received previously from other nodes:

$$m_{\alpha \to \beta}(\tau_{\beta}) \qquad (2)$$

$$\propto \sum_{\tau_{\alpha} \in \{0,1\}} n_{\alpha \to \beta}(\tau_{\alpha}) \psi_{\alpha}(\tau_{\alpha}) \psi_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}),$$

where

$$n_{\alpha \to \beta}(\tau_{\alpha}) \stackrel{\text{def}}{=} \prod_{\gamma \in V(\alpha) \setminus \{\beta\}} m_{\gamma \to \alpha}(\tau_{\alpha}), \tag{3}$$

and the proportionality symbol \propto indicates that one must normalise the messages so that they sum to 1. The beliefs b_{α} are then reconstructed according to

$$b_{\alpha}(\tau_{\alpha}) \propto \psi_{\alpha}(\tau_{\alpha}) \prod_{\beta \in V(\alpha)} m_{\beta \to \alpha}(\tau_{\alpha}),$$
 (4)

and similarly, the belief $b_{\alpha\beta}$ of the joint probability of $(\tau_{\alpha}, \tau_{\beta})$ is given by

$$b_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}) \propto n_{\alpha \to \beta}(\tau_{\alpha}) n_{\beta \to \alpha}(\tau_{\beta})$$

$$\times \psi_{\alpha}(\tau_{\alpha}) \psi_{\beta}(\tau_{\beta}) \psi_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}).$$
(5)

A simple computation shows that equations (4) and (5) are compatible, since (2)–(3) imply that

$$\sum_{\tau_{\alpha} \in \{0,1\}} b_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}) = b_{\beta}(\tau_{\beta}).$$

We propose to use the BP algorithm for two purposes: estimation of the model parameters (the functions ψ) from historical data and reconstruction of traffic from current data.

Estimation of the model

The fixed points of the BP algorithm (and therefore the Bethe approximation) allows to approximate the joint marginal probability $p_{\alpha\beta}$ when the functions $\psi_{\alpha\beta}$ are known. Conversely, it can provide good candidates for $\psi_{\alpha\beta}$ from the historical values $\hat{p}_{\alpha\beta}$ and \hat{p}_{α} .

We are looking for a fixed point of the BP algorithm satisfying (2)-(3) and such that $b_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}) = \hat{p}_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta})$ and therefore $b_{\alpha}(\tau_{\alpha}) = \hat{p}_{\alpha}(\tau_{\alpha})$.

It is easy to check that the following choice of ψ admits a fixed point (constant messages equal to 1/2) with the correct beliefs.

$$\psi_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta}) = \frac{\hat{p}_{\alpha\beta}(\tau_{\alpha}, \tau_{\beta})}{\hat{p}_{\alpha}(\tau_{\alpha})\hat{p}_{\beta}(\tau_{\beta})}, \qquad (6)$$

$$\psi_{\alpha}(\tau_{\alpha}) = \hat{p}_{\alpha}(\tau_{\alpha}). \qquad (7)$$

$$\psi_{\alpha}(\tau_{\alpha}) = \hat{p}_{\alpha}(\tau_{\alpha}). \tag{7}$$

There is no guarantee however that the fixed points will be attained by the BP algorithm, so the choice (6)–(7) functions is only heuristic.

Traffic reconstruction and prediction

Let \mathcal{N}^* be the set of nodes that have been visited by probe vehicles. The problem of reconstructing traffic from the data gathered by those vehicles is equivalent to evaluating the conditional probability

$$p_{\alpha}(\tau_{\alpha}|\tau_{\mathcal{N}^*}) = \frac{p_{\alpha,\mathcal{N}^*}(\tau_{\alpha},\tau_{\mathcal{N}^*})}{p_{\mathcal{N}^*}(\tau_{\mathcal{N}^*})},$$

where $\tau_{\mathcal{N}^*}$ is a shorthand notation for the set $\{\tau_{\alpha^*}\}_{\alpha^* \in \mathcal{N}^*}$.

The BP algorithm applies to this case if a specific rule is defined for nodes $\alpha^* \in \mathcal{N}^*$: since the value of τ_{α^*} is known, there is no need to sum over possible values and

$$m_{\alpha^* \to \beta}(\tau_{\beta}) = n_{\alpha^* \to \beta}(\tau_{\alpha^*}) \psi_{\alpha^*}(\tau_{\alpha^*}) \psi_{\alpha^* \beta}(\tau_{\alpha^*}, \tau_{\beta}).$$

The resulting algorithm is supposed to be run in real time, over a graph which correspond to a time window (typically a few hours) centred around present time, with probe vehicle data added as it is available. In this perspective, the reconstruction and prediction operations are done simultaneously on an equal footing, the distinction being simply the time-stamp (past for reconstruction or future for prediction) of a given computed belief. The output of the previous run can be used as initial messages for a new run, in order to speedup convergence. Full re-initialisation (typically a random set of initial messages) has to be performed within a time interval of the order but smaller than the time-scale of typical traffic fluctuations.

Conclusion

The algorithm presented above has been implemented and tested using a traffic toy model which is essentially a queueing system exhibiting two basic traffic patterns. We have found that the BP fixed point corresponding to the heuristic of Section is unstable; it indeed encodes a superposition of two stable fixed points corresponding to each traffic pattern. Actually, no information is lost in this procedure: our simulations indicate indeed that a few probe vehicles (less than 3 for a network consisting of 60 links) are enough to detect the time dependent state of the traffic and perform a faithful reconstruction. It is used in the REACT project for the final demonstration taking place in September 2006 in Munich. However, a preliminary experiment in June demonstrated that it was not sufficiently stable to ensure good predictions. This is obviously due to the lack of data: only 5 cars have been fully equipped. INRIA and Mines Paris are now continuing the research effort on that topic on their own, in collaboration with French road authorities to get more real data. This algorithm seems promising and raised the interest of several private companies.

The Bibliography

- 1. T. Benz et al., Information supply for intelligent routing services the INVENT traffic network equalizer approach, Proceedings of the ITS World Congress (2003).
- 2. J. Essen and M. Schreckenberg, Microscopic simulation of urban traffic based on cellular automata, Int. J. Mod. Phys. (1997), C8:1025–1036.
 - 3. E Ising, Beitrag zur theory des ferromagnetismus, Zeitschr. f. Phys. 31 (1925).
- 4. J. M. Mooij and H. J. Kappen, On the properties of the bethe approximation and loopy belief propagation on binary network, J. Stat. Mech. (2005), P11012.

- M. Mzard, G. Parisi, and M.A. Virasoro, Spin glass theory and beyond, World Scientific, Singapore, 1987.
- K. Nagel and M. Schreckenberg, A cellular automaton model for freeway traffic, J. Phys. I,2 (1992), 2221–2229.
- 7. J. Pearl, Probabilistic reasoning in intelligent systems: Network of plausible inference, Morgan Kaufmann, 1988.
 - 8.PRIME project, Technology assessment and expected targets, Deliverable D3.2, 2000.
- 9. H. H. Versteegt and C. M. J. Tampre, *Predictime state of the art and functional architecture*, TNO Inro report 2003-07 (2003).
- 10. J. S. Yedidia, Freeman W. T., and Y. Weiss, Constructing free-energy approximations and generalized belief propagation algorithms, IEEE Trans. Inform. Theory. **51** (2005), no. 7, 2282–2312.

Bounds and algebraic algorithms in differential algebra: the ordinary case ³

Oleg Golubitsky,

Ontario Research Centre for Computer Algebra, University of Western Ontario, London, Ontario, Canada, N6A 5B7, E-mail: oleg.golubitsky@gmail.com

Marina V. Kondratieva,

Department of Mechanics and Mathematics,
Moscow State University, Leninskie gory, Moscow, Russia, 119 992,
E-mail: kondra_m@shade.msu.ru

Marc Moreno Maza,

Ontario Research Centre for Computer Algebra, University of Western Ontario, London, Ontario, Canada, N6A 5B7, E-mail: moreno@orcca.on.ca

Alexey Ovchinnikov,

Department of Mathematics,
North Carolina State University, Raleigh, North Carolina, USA, 27695-8205,
E-mail: aiovchin@ncsu.edu

Consider the Rosenfeld-Groebner algorithm for computing a regular decomposition of a radical differential ideal. We propose a bound on the orders of derivatives occurring in all intermediate and final systems computed by this algorithm. We also reduce the problem of conversion of a regular decomposition of a radical differential ideal from one ranking to another to a purely algebraic problem.

Keywords: differential algebra, characteristic sets, radical differential ideals, regular decomposition.

Introduction

Consider the ring of ordinary differential polynomials $\mathbf{k}\{Y\}$, where \mathbf{k} is a differential field of characteristic 0 with derivation δ , and $Y = \{y_1, \dots, y_n\}$ is a set whose elements are called differential indeterminates. Let $F \subset \mathbf{k}\{Y\}$ be a set of differential polynomials, then [F] and $\{F\}$ denote the differential and radical differential ideals generated by F in $\mathbf{k}\{Y\}$, respectively. A differential ideal may not have a finite generating system, while a radical differential ideal always has one according to the Basis Theorem [13]. One of the central problems in constructive differential algebra is the problem of computing a canonical representation for a radical differential ideal.

The problem, in general, remains open, but an important contribution to it is provided by the Rosenfeld-Gröbner algorithm [2]. This algorithm inputs a set of differential polynomials F and a ranking

³The work has been supported by the Russian Foundation for Basic Research, project no. 05-01-00671, by NSF Grant CCR-0096842, by NSERC Grant PDF-301108-2004, and by NSERC Grant RGPIN Algorithms and software for triangular decompositions of algebraic and differential systems.

[9] on the set of derivatives of the indeterminates. By applying differential pseudo-reductions [13,9] to the elements of F and considering their initials and separants H_F (these operations depend on the ranking), the algorithm constructs finitely many systems of the form $F_i = 0$, $H_i \neq 0$, where $F_i, H_i \subset \mathbf{k}\{Y\}$, $i = 1, \ldots, m$. At any intermediate step of the algorithm, these systems are equivalent to F: each solution of F = 0 is a solution of $F_i = 0$, $H_i \neq 0$ for some i and vice versa. The algorithm terminates when all systems $F_i = 0$, $H_i \neq 0$ become regular [2]. The resulting regular decomposition $\{F\} = \bigcap_{i=1}^m [F_i] : H_i^{\infty}$ solves the membership problem for $\{F\}$ [2]: $f \in \{F\}$ iff the differential pseudo-remainder of f w.r.t. F_i belongs to the algebraic ideal $(F_i) : H_i^{\infty}$, for all $i \in \{1, \ldots, m\}$.

Computational complexity of the Rosenfeld-Gröbner algorithm is an open problem. Yet for the corresponding algebraic problem of computing a regular decomposition of a radical algebraic ideal in $\mathbf{k}[Y]$, bounds on complexity are known [15]. Thus, the first natural step towards obtaining complexity bounds in the differential case would be estimating the orders of derivatives occurring in the polynomials computed by the Rosenfeld-Gröbner algorithm. For systems of linear differential polynomials and systems of two differential polynomials in two indeterminates, Ritt [12] has proved that the Jacobi bound on the orders holds. The Rosenfeld-Gröbner algorithm was discovered later, but Ritt's techniques provide the starting point for our analysis of this algorithm.

Bound on the orders of derivatives

Our first result provides a bound for the orders of derivatives occurring in the systems $F_i = 0$, $H_i \neq 0$ (for an arbitrary ranking). Let $m_i(F)$ be the maximal order of a derivative of the *i*-th indeterminate occurring in F, and let

$$M(F) = \sum_{i=1}^{n} m_i(F).$$

We propose a modification of the Rosenfeld-Gröbner algorithm, in which for every intermediate system $F_i = 0, H_i \neq 0$, we have

$$M(F_i \cup H_i) \le (n-1)!M(F).$$

Given a set F of differential polynomials and a ranking, the conventional Rosenfeld-Gröbner algorithm at first computes a characteristic set $\mathbb C$ of F, i.e., an autoreduced subset of F of the least rank. We replace this computation by that of a weak d-triangular subset of F of the least rank, which we call a weak characteristic set of F. A set $\mathbb C \subset \mathbf k\{Y\} \setminus \mathbf k$ is called a weak d-triangular set [8,Definition 3.7], if the set of its leaders $\operatorname{Id} \mathbb C$ is autoreduced. In the ordinary case, $\mathbb C$ is a weak d-triangular set if and only if the leading differential indeterminates $\operatorname{Iv} f, f \in \mathbb C$, are all distinct. The differential pseudo-remainder of a polynomial f w.r.t. a weak d-triangular set $\mathbb C$ is defined via [8,Algorithm 3.13]. Weak characteristic sets satisfy the following property essential for the proof of our bound:

Lemma 1. Let F be a set of differential polynomials, and let \mathbb{C} be a weak characteristic set of F. Then $lv \mathbb{C} = lv F$.

Second, the Rosenfeld-Gröbner algorithm computes the differential pseudo-remainders of $F \setminus \mathbb{C}$ w.r.t. \mathbb{C} . The orders of derivatives of non-leading indeterminates (i.e., those not in $\operatorname{lv}\mathbb{C}$) occurring in these pseudo-remainders may be higher than those in F (unless the chosen ranking is orderly). In order to control this growth of orders, we construct a differential prolongation of the weak characteristic set \mathbb{C} , i.e., an algebraically triangular set \mathbb{B} such that the differential pseudo-reduction of $F \setminus \mathbb{C}$ w.r.t. \mathbb{C} can be replaced by the algebraic pseudo-reduction w.r.t. \mathbb{B} . We give the specification of the algorithm computing the differential prolongation, leaving out the details of the computation:

```
 \begin{aligned} \textbf{Algorithm} & \text{ Differentiate\&Autoreduce}(\mathbb{C}, \{m_i\}) \\ & \text{ Input:} & \text{ a weak d-triangular set } \mathbb{C} = C_1, \dots, C_k \\ & \text{ with } \text{ ld } \mathbb{C} = y_1^{(d_1)}, \dots, y_k^{(d_k)}, \\ & \text{ and a set of non-negative integers } \{m_i\}_{i=1}^k, \ m_i \geq m_i(\mathbb{C}) \\ & \text{ Output:} & \text{ set } \mathbb{B} = \{B_i^j \mid 1 \leq i \leq k, \ 0 \leq j \leq m_i - d_i\} \text{ satisfying} \\ & \text{ rk } B_i^j = \text{rk } C_i^{(j)} \\ & \mathbb{B} \subset [\mathbb{B}^0] \subset [\mathbb{C}] \subset [\mathbb{B}] : H_{\mathbb{B}}^{\infty}, \text{ where} \\ & \mathbb{B}^0 = \{B_i^0 \mid 1 \leq i \leq k\} \\ & H_{\mathbb{B}} \subset H_{\mathbb{C}}^{\infty} + [\mathbb{C}], \ H_{\mathbb{B}}^{\infty} H_{\mathbb{C}} \subset H_{\mathbb{B}}^{\infty} + [\mathbb{B}] \\ & B_i^j \text{ are partially reduced w.r.t. } \mathbb{C} \setminus \{C_i\} \\ & m_i(\mathbb{B}) \leq m_i(\mathbb{C}) + \sum_{j=1}^k (m_j - d_j), \ i = k+1, \dots, n \\ & \text{ or } \{1\}, \text{ if it is detected that } [\mathbb{C}] : H_{\mathbb{C}}^{\infty} = (1) \end{aligned}
```

We obtain the following modification of the Rosenfeld-Gröbner algorithm:

```
Algorithm RGBound(F_0, H_0)
  INPUT:
                          sets of differential polynomials F_0, H_0
                         a set T of regular systems such that \{F_0\}: H_0^{\infty} = \bigcap_{(\mathbb{A},H)\in T} [\mathbb{A}]: H^{\infty},
M(\mathbb{A} \cup H) \leq (n-1)! M(F_0 \cup H_0) \text{ for } (\mathbb{A},H) \in T.
  OUTPUT:
         T := \varnothing, \quad U := \{(F_0, \varnothing, H_0)\}
         while U \neq \emptyset do
                 Take and remove any (F, \mathbb{C}, H) \in U
                 f := an element of F of the least rank
                D := \{ C \in \mathbb{C} \mid \operatorname{lv} C = \operatorname{lv} f \}
                G := F \cup D \setminus \{f\}
                \overline{\mathbb{C}} := \mathbb{C} \setminus D \cup \{f\}
                \mathbb{B} := \mathsf{Differentiate\&Autoreduce}(\bar{\mathbb{C}}, \{m_i(G \cup \bar{\mathbb{C}} \cup H) \mid y_i \in \mathsf{lv}\,\bar{\mathbb{C}}\})
                if \mathbb{B} \neq \{1\} then
                        \bar{F} := \mathsf{algrem}(G, \mathbb{B}) \setminus \{0\}
                        \bar{H} := \mathsf{algrem}(H, \mathbb{B}) \cup H_{\mathbb{R}}
                       if \bar{F} \cap \mathbf{k} = \emptyset and 0 \notin \bar{H} then
                              if \bar{F} = \emptyset then T := T \cup \{(\mathbb{B}^0, \bar{H})\}
                              else U := U \cup \{(\bar{F}, \bar{\mathbb{C}}, \bar{H})\}
                 U := U \cup \{ (F \cup \{h\}, \mathbb{C}, H) \mid h \in H_f \setminus K \}
         end while
         return T
```

Algebraic conversion of characteristic sets

Our second result is a reduction of the problem of conversion of a regular decomposition of a radical differential ideal from one ranking to another to a purely algebraic problem. For the algebraic case, efficient modular algorithms are currently being developed [4] and implemented using the RegularChains library in Maple [10]; a parallel implementation on a shared memory machine in Aldor is also in progress [11].

We note that each regular component $[F_i]: H_i^{\infty}$ can be decomposed further into an intersection of characterizable differential ideals [7] of the form $I_j = [\mathbb{C}_j]: H_{\mathbb{C}_j}^{\infty}$, where \mathbb{C}_j is an autoreduced subset of I_j of the least rank (called a characteristic set [9] of I_j). Then we obtain a characteristic decomposition $\{F\} = \bigcap_{j=1}^t I_j$ of the radical differential ideal.

A prime differential ideal I is characterizable w.r.t. any ranking, and for any characteristic set $\mathbb C$ of I, we have $I=[\mathbb C]:H^\infty_{\mathbb C}$. The minimal differential prime components (called the essential prime components) of a characterizable ideal $I=[\mathbb C]:H^\infty_{\mathbb C}$ correspond to the minimal prime components of the algebraic ideal $(\mathbb C):H^\infty_C$ [7]: an autoreduced set $\mathbb A$ is a characteristic set of a minimal prime of $(\mathbb C):H^\infty_{\mathbb C}$ if and only if $\mathbb A$ is a characteristic set of an essential prime component of I; the corresponding algebraic and differential prime components are equal to $(\mathbb A):H^\infty_{\mathbb A}$ and $[\mathbb A]:H^\infty_{\mathbb A}$, respectively. Moreover, the leading derivatives of $\mathbb A$ coincide with those of $\mathbb C$.

We first consider a **special case**, when the given characterizable ideal $I = [\mathbb{C}] : H^{\infty}_{\mathbb{C}}$ is prime, and it is required to convert its characteristic set \mathbb{C} from one ranking to another (the problem of efficient conversion of characteristic sets of prime differential ideals from one ranking to another has been addressed in [1,3,5]).

Given the orders of derivatives occurring in \mathbb{C} , we provide a bound on the orders of derivatives occurring in a characteristic set of I w.r.t. the target ranking. Based on [14,Theorem 24] (if the target ranking is an elimination ranking) or [6,Theorem 6] (for an arbitrary target ranking), we can show that a bound of $n \cdot \max m_i(\mathbb{C})$ holds.

Using this bound, we find a prime algebraic sub-ideal $J \subset I$, which contains a characteristic set $\bar{\mathbb{C}}$ of I w.r.t. the target ranking. Then we compute the canonical algebraic characteristic set of J w.r.t. the target ranking and extract from it the canonical characteristic set of I.

We have carried out a preliminary implementation of this algorithm in Maple, using the RegularChains library.

Now consider the **general case**, when we are given an arbitrary characterizable differential ideal $I = [\mathbb{C}] : H^{\infty}_{\mathbb{C}}$ and need to compute its characteristic decomposition w.r.t. another ranking. Since the essential prime components of I correspond to the minimal primes of the algebraic ideal $(\mathbb{C}) : H^{\infty}_{\mathbb{C}}$, and

thus their characteristic sets can be computed from \mathbb{C} without applying differentiations, we have the bound $M = n \cdot \max m_i(\mathbb{C})$ for the characteristic sets of the essential primes of I w.r.t. the target ranking.

Let $d = \max_{f \in \mathbb{C}} (M - \text{ord } \text{ld } f)$, where ld f denotes the leading derivative of f w.r.t. the initial ranking and ord ld f is its order, and let

$$\mathbb{C}^{(d)} = \{ f^{(k)} \mid f \in \mathbb{C}, \ 0 \le k \le d \}.$$

Applying a purely algebraic (and factorization-free) algorithm to the ideal $J=(\mathbb{C}^{(d)}):H^\infty_{\mathbb{C}}$, we compute its decomposition $J'_1\cap\ldots\cap J'_l$ into algebraic "bi-characterizable" components, i.e., ideals characterizable w.r.t. both initial and target rankings.

We observe that a component J'_i , whose characteristic set w.r.t. the initial ranking has a set of leaders distinct from $\operatorname{ld} \mathbb{C}^{(d)}$, is a redundant component, i.e., $J = \cap_{j \neq i} J'_j$. So, we can assume that the characteristic sets of J'_i have leaders equal to $\operatorname{ld} \mathbb{C}^{(d)}$ for all $i = 1, \ldots, l$. We prove then that every minimal prime component Q of J'_i is also a minimal prime component of J, hence it corresponds to an essential prime component P of I.

Now, due to the choice of d, every minimal prime of $J = (\mathbb{C}^{(d)}) : H_{\mathbb{C}}^{\infty}$ contains a differential characteristic set of the corresponding essential prime of I w.r.t. any ranking. We take the canonical algebraic characteristic set of J'_i w.r.t. the target ranking and extract from it the canonical characteristic set \mathbb{B}_i of I'_i . Since the essential primes of I'_i are those essential primes of I that contain the minimal primes of J'_i , we obtain a characteristic decomposition w.r.t. the target ranking:

$$I = \bigcap_{i=1}^{l} I_i' = \bigcap_{i=1}^{l} [\mathbb{B}_i] : H_{\mathbb{B}_i}^{\infty}.$$

The Bibliography

- 1. F. Boulier. Efficient computation of regular differential systems by change of rankings using Kähler differentials. Technical report, Université Lille, 1999.
- 2. F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Representation for the radical of a finitely generated differential ideal. In *Proc. ISSAC'95*, pages 158–166. ACM Press, 1995.
 - 3. F. Boulier, F. Lemaire, and M. Moreno Maza. PARDI! Technical report, LIFL, 2001.
- 4. X. Dahan, X. Jin, M. Moreno Maza, and É Schost. Change of ordering for regular chains in positive dimension. In Ilias S. Kotsireas, editor, Maple Conference 2006, pages 26–43, 2006.
- 5. O. Golubitsky. Gröbner walk for characteristic sets of prime differential ideals. In V. Ganzha, E. Mayr, and E. Vorozhtsov, editors, *Proc. 7th Workshop on Comp. Alg. in Sc. Comp.*, pages 207–221. TU München, Germany, 2004.
- O. Golubitsky, M. Kondratieva, and A. Ovchinnikov. Canonical characteristic sets of characterizable differential ideals. *Preprint*, 2005.
- 7. E. Hubert. Factorization-free decomposition algorithms in differential algebra. *J. Symb. Comput.*, 29:641–662, 2000.
- 8. E. Hubert. Notes on triangular sets and triangulation-decomposition algorithms II: Differential systems. In *Symbolic and Numerical Scientific Computing 2001*, pages 40–87, 2003.
 - 9. E.R. Kolchin. Differential Algebra and Algebraic Groups. Academic Press, 1973.
- 10. F. Lemaire, M. Moreno Maza, and Y. Xie. The RegularChains library. In Ilias S. Kotsireas, editor, Maple Conference 2005, pages 355–368, 2005.
- 11. M. Moreno Maza and Y. Xie. Parallelization of triangular decomposition. In *Proc. of Algebraic Geometry and Geometric Modeling'06*, Spain, 2006. University of Barcelona.
- 12. J.F. Ritt. Jacobi's problem on the order of a system of differential equations. The Annals of Mathematics, 36(2):303-312, 1935.

- 13. J.F. Ritt. Differential Algebra. Dover, 1950.
- 14. B. Sadik. A bound for the order of characteristic set elements of an ordinary prime differential ideal and some applications. Applicable Algebra in Engineering, Communication and Computing, 10(3):251–268, 2000.
 - 15. Á. Szántó. Computation with Polynomial Systems. PhD thesis, Cornell University, 1999.

No trades under rationality about expectations ⁴

Takashi Matsuhisa

Department of Natural Sciences, Ibaraki National College of Technology, 866 Nakane, Hitachinaka-shi, Ibaraki 312-8508, Japan. Tel: +81-29-271-2865; fax: +81-29-271-2857. E-mail: mathisa@ge.ibaraki-ct.ac.jp

Ryuichiro Ishikawa,

Graduate School of Systems and Information Engineering,
University of Tsukuba,
1-1-1, Ten-nodai, Tsukuba, Ibaraki 305-8573, Japan.
E-mail: ishikawa@sk.tsukuba.ac.jp

We reconsider the no trade theorem in an exchange economy where the traders have non-partition information from aspect of rough sets. By introducing a new concept, *rationality of expectations*, we show some versions of the theorem different from previous works, such as Geanakoplos (1989).

Keywords: No trade theorem, Ex ante Pareto optimum, Common knowledge, Rational expectations equilibrium.

Introduction

The no trade theorem has shown that new information will not give the traders any incentive to trade when their initial endowments are allocated ex ante Pareto-optimally. In this theorem, there is the standard assumption: the partitional information structure. This paper explores the extent to which this assumptions is generalized in the theorem.

In recent years, several investigators have already generalized the assumptions in this theorem. Geanakoplos (1989) neatly analyzes non-partition information structure With this concept, he examines several classes of non-partition information and the relations among them, and characterizes Nash equilibrium and rational expectations equilibrium in those classes.

Our paper discusses similar issues, but captures different features from his analysis with a new concept: rationality of expectations. This concept means that each trader knows his own expected utility. As shown later, this requirement does not necessarily imply either partitional information structure or positive balancedness. Moreover it does not require that traders are risk-neutral or risk-averse, which is usually assumed in this literature (c.f. Milgrom and Stokey (1982), and Sebenius and Geanakoplos (1983)).

Several variations of the no trade theorem have been developed. Neeman (1996) applies it in the case of p-beliefs, Luo and Ma (2000) in the non-expected utility case, Morris and Skiadas (2000) in the case of rationalizable trades, and so on. Our model applies it to expected utility and rational expectations equilibrium, and therefore uses the standard setting of the original as Milgrom and Stokey (1982) and Sebenius and Geanakoplos (1983).

This paper is organized as follows: In Section 1 we define an economy with non-partition information structure and rational expectations equilibrium in our economy. The key notion, rationality of expectations, is defined in this section. In Section 2 we show two extended no trade theorems, and we comment on welfare of the rational expectations equilibrium in our economy. In Section 3, we give an example to compare with Geanakoplos (1989). In the example, we consider non-partition information different from that of Geanakoplos, and we give comments on the common prior assumption.

 $^{^4}$ Partially supported by Grants-in-Aid for Scientific Research(C)(2)(No.18540153) from the Japan Society for the Promotion of Sciences

1. Model of an exchange economy

Let Ω be a non-empty *finite* set called a *state space* and let 2^{Ω} denote the field of all subsets of Ω . Each member of 2^{Ω} is called an *event* and each element of Ω called a *state*. We consider the set N of n traders; i.e., $N = \{1, 2, \dots, n\}$.

Information and Knowledge

We define i's knowledge operator $K_i: 2^{\Omega} \to 2^{\Omega}$. To interpret $K_i(E)$ as the event that i knows E, we specify that K_i satisfies the postulates 'Truth' **T** and 'positive introspection' **4**:

T $K_i E \subseteq E$ for every $E \in 2^{\Omega}$. **4**: $K_i E \subseteq K_i K_i E$ for every $E \in 2^{\Omega}$.

i's knowledge operator K_i gives rise to i's possible operator $P_i: 2^{\Omega} \to 2^{\Omega}$ defined by $P_i(E) := \Omega \setminus K_i(\Omega \setminus E)$, and i's possibility correspondence $P_i: \Omega \to 2^{\Omega}$ defined by $P_i(\omega) := P_i(\{\omega\})$, which satisfies the following two conditions for every $i \in N$:

Ref : $\omega \in P_i(\omega)$ for every $\omega \in \Omega$.

Trn: $\xi \in P_i(\omega)$ implies $P_i(\xi) \subseteq P_i(\omega)$ for all $\xi, \omega \in \Omega$.

The class of correspondences $(P_i)_{i\in N}$ is called RT-information structure.⁵ The event $P_i(\omega)$ is interpreted as the set of all the states that trader i thinks are possible at ω . Note that the correspondence P_i gives an information consisting of rough sets because $K_i(P_i(\omega)) \neq P_i(\omega)$. The common knowledge operator K_C is defined by the infinite recursion of knowledge operators: $K_CE := \bigcap_{k=1,2,\ldots} \bigcap_{\{i_1,i_2,\ldots,i_k\}\subset N} K_{i_1}K_{i_2}\cdots K_{i_k}E$. Given the RT-information structure $(P_i)_{i\in N}$, the commonly possible operator is the correspondence $M:\Omega\to 2^\Omega$ defined by $M(\omega)=\bigcup (P_{i_1}(P_{i_2}(\cdots P_{i_k}(\omega)\cdots)))$, where the union ranges over all finite sequences of traders. We note that $\omega\in K_CE$ if and only if $M(\omega)\subseteq E$.

Economy with RT-information structure

We define a pure exchange economy with RT-information structure \mathcal{E} as a tuple $\langle N, (\Omega, (P_i, \mu_i)_{i \in N}), (e_i, U_i)_{i \in N} \rangle$, which consists of the following structure and interpretations: There are l commodities at each state, and it is assumed that i's consumption set at each state is \mathbb{R}^l_+ . Each trader i has a state-dependent endowment $e_i : \Omega \to \mathbb{R}^l_+$ with $\sum_{i \in N} e_i(\omega) > 0$ for all $\omega \in \Omega$, a quasi-concave von Neumann-Morgenstern utility function $U_i : \mathbb{R}^l_+ \times \Omega \to \mathbb{R}$, and a subjective prior μ_i on Ω with full support⁶ for every $i \in N$. In our economy \mathcal{E} , we assume that i's utility function $U_i(\cdot, \omega)$ for each ω is continuous and strictly quasi-concave.

The traders trade according to a profile $\mathbf{t} = (t_i)_{i \in N}$ of functions t_i from Ω into \mathbb{R}^l . A trade is said to be feasible if, for all $i \in N$ and for all $\omega \in \Omega$, $e_i(\omega) + t_i(\omega) \geq 0$ and $\sum_{i \in N} t_i(\omega) \leq 0$. Given initial endowments $(e_i)_{i \in N}$ and any feasible trade $\mathbf{t} = (t_i)_{i \in N}$, we refer to $(e_i + t_i)_{i \in N}$ as an allocation $\mathbf{a} = (a_i)_{i \in N}$. Note that an allocation is $\sum_{i \in N} a_i(\omega) \leq \sum_{i \in N} e_i(\omega)$ for every $\omega \in \Omega$. We denote by \mathcal{A} the set of all allocations and denote by \mathcal{A}_i the set of all the i's components: $\mathcal{A} = \times_{i \in N} \mathcal{A}_i$. For i's allocation $a_i \in \mathcal{A}_i$, each trader i has expectations; i's ex ante expectation is defined by $\mathbf{E}_i[U_i(a_i)] := \sum_{\omega \in \Omega} U_i(a_i(\omega), \omega) \mu_i(\omega)$. The endowments $(e_i)_{i \in N}$ are said to be ex ante Pareto-optimal if there is no allocation $(a_i)_{i \in N}$ such that $\mathbf{E}_i[U_i(a_i)] \geq \mathbf{E}_i[U_i(e_i)]$ for every trader $i \in N$ with at least one strict inequality.

For i's allocation $a_i \in \mathcal{A}_i$, we define i's interim expectation at $\omega \in \Omega$ as $\mathbf{E}_i[U_i(a_i)|P_i](\omega) := \sum_{\xi \in \Omega} U_i(a_i(\xi), \xi) \mu_i(\xi|P_i(\omega))$. Given a feasible trade $\mathbf{t} = (t_i)_{i \in N}$, t_i is acceptable for trader $i \in N$ at state $\omega \in \Omega$ if $\mathbf{E}_i[U_i(e_i+t_i)|P_i](\omega) \geq \mathbf{E}_i[U_i(e_i)|P_i](\omega)$. We denote by $Acp(t_i)$ the set of all the states in which t_i is acceptable for i, and denote $Acp(\mathbf{t}) := \bigcap_{i \in N} Acp(t_i)$. Furthermore we set the event of i's interim expectation for the trade t_i at ω :

$$[\mathbf{E}_{i}[U_{i}(e_{i}+t_{i})|P_{i}](\omega)] := \{\xi \in \Omega \mid \mathbf{E}_{i}[U_{i}(e_{i}+t_{i})|P_{i}](\xi) = \mathbf{E}_{i}[U_{i}(e_{i}+t_{i})|P_{i}](\omega)\}.$$

Given the event $[\mathbf{E}_i[U_i(e_i+t_i)|P_i](\omega)]$, we denote $R_i(t_i) = \{\omega \in \Omega \mid P_i(\omega) \subseteq [\mathbf{E}_i[U_i(e_i+t_i)|P_i](\omega)] \}$ and $R(t) = \bigcap_{i \in N} R_i(t_i)$.

Definition 1. A trader i is rational about his expectation for his trade t_i at ω if $\omega \in R_i(t_i)$; that is, $\omega \in K_i([\mathbf{E}_i[U_i(e_i+t_i)|P_i](\omega)))$. A trader i is rational everywhere about his expectation for t_i if $R_i = \Omega$.

The event $R_i(t_i)$ means that trader i knows his expected gain from t_i at ω . Trader i is interpreted as knowing his interim expected utility at ω . If we consider the standard information structure of a partition on Ω , trader i is necessarily rational everywhere; i.e., $R_i(t_i) = \Omega$.

 $^{{}^5{}m The}\ RT$ -information structure stands for the reflexive and transitive information structure.

⁶I.e., $\mu_i(\omega) > 0$ for every $\omega \in \Omega$.

Price system and rational expectations equilibrium

A price system is a positive function $p: \Omega \to \mathbb{R}^l_{++}$. The budget set of a trader i at a state ω for a price system p is defined by $B_i(\omega,p) = \{ a \in \mathbb{R}^l_+ \mid p(\omega) \cdot a \leq p(\omega) \cdot e_i(\omega) \}$. We denote $\Delta(p)(\omega) := \{ \xi \in \Omega \mid p(\xi) = p(\omega) \}$ and $\Delta(p)$ the partition induced by p; i.e., $\Delta(p) = \{ \Delta(p)(\omega) \mid \omega \in \Omega \}$. When trader i learns from prices, his new information is represented by a mapping $\Delta(p) \cap P_i : \Omega \to 2^{\Omega}$ defined by $(\Delta(p) \cap P_i)(\omega) := \Delta(p)(\omega) \cap P_i(\omega)$.

Definition 2 (Geanakoplos (1989)). A rational expectations equilibrium for an economy \mathcal{E} is a pair (p, \mathbf{x}) , in which p is a price system and $\mathbf{x} = (x_i)_{i \in N}$ is an allocation satisfying the following conditions:

- **RE 1** For every $\omega \in \Omega$, $\sum_{i \in N} x_i(\omega) = \sum_{i \in N} e_i(\omega)$.
- **RE 2** For every $\omega \in \Omega$ and each $i \in N$, $x_i(\omega) \in B_i(\omega, p)$.
- **RE 3** If $P_i(\omega) = P_i(\xi)$ and $p(\omega) = p(\xi)$, then $x_i(\omega) = x_i(\xi)$ for trader $i \in N$ for any $\xi, \omega \in \Omega$.
- **RE 4** For each $i \in N$ and any mapping $y_i : \Omega \to \mathbb{R}^l_+$ with $y_i(\omega) \in B_i(\omega, p)$ for all $\omega \in \Omega$, $\mathbf{E}_i[U_i(x_i)|\Delta(p)\cap P_i](\omega) \geqq \mathbf{E}_i[U_i(y_i)|\Delta(p)\cap P_i](\omega)$.

The profile $\mathbf{x} = (x_i)_{i \in \mathbb{N}}$ is called a rational expectations equilibrium allocation.

For i's trade t_i , we set $R_i(p,t_i) := \{\omega \in \Omega | (\Delta(p) \cap P_i)(\omega) \subseteq [\mathbf{E}_i[U_i(e_i+t_i)|\Delta(p) \cap P_i](\omega)]\}$, and denote $R(p,t) = \bigcap_{i \in N} R_i(p,t_i)$. The set $R_i(p,t_i)$ is interpreted as the event that i knows his i interim expectation for his trade t_i when he receives some new information from the price system p, and R(p,t) is interpreted as the event that everyone knows his interim expectation for his trade with the price system p.

Definition 3. A trader i is said to be rational about his expectation for t_i with a price system p at ω if $\omega \in R_i(p, t_i)$. All traders are rational everywhere about their expectations for t with p if $R(p, t) = \Omega$.

2. No trade theorems

In this section we shall give two extensions of the no trade theorem of Milgrom and Stokey (1982). In addition, we show the welfare of the rational expectations equilibrium.

No trade theorem with RT-information structure

The following is a direct extension of Milgrom and Stokey's theorem to an economy with RT-information structure, which will be proved in Appendix.

Theorem 1. Let \mathcal{E} be an economy with RT-information structure, and let $\mathbf{t} = (t_i)_{i \in \mathbb{N}}$ be a feasible trade. Suppose that the initial endowments $(e_i)_{i \in \mathbb{N}}$ are ex ante Pareto optimal. Then the traders can never agree to any non-null trade at each state where they commonly know both the acceptable trade $\mathbf{t} = (t_i)$ and where they are rational about their expectations for the trade; that is, $\mathbf{t}(\omega) = \mathbf{0}$ at every $\omega \in K_C(Acp(\mathbf{t}) \cap R(\mathbf{t}))$.

Welfare in an economy with knowledge

We examine the welfare of the rational expectations equilibrium in our economy. It is characterized from the viewpoint of ex ante optimality. This will be proved in Appendix as well as Theorem 1.

Proposition 1. In an economy with RT-information structure \mathcal{E} , let an allocation $\mathbf{x} = (x_i)_{i \in N}$ be a rational expectations equilibrium allocation relative to some price system p with which all the traders are rational everywhere about their expectations with respect to $(x_i - e_i)_{i \in N}$. Then \mathbf{x} is ex ante Pareto optimal.

3. Concluding remarks

In our model, we impose reflexivity and transitivity on traders' information structure while Geanakoplos imposes reflexivity and positive balancedness as follows: The information structure (Ω, P) is called *positively balanced* with respect to $E \subset \Omega$ if there is a function $\lambda : \underline{P} \to \mathbb{R}_+$ such that $\sum_{\substack{C \subseteq \underline{P} \\ C \subset E}} \lambda(C) \chi_C(\omega) = \chi_E$ for all $\omega \in \Omega$, where $\underline{P} := \{F \in 2^{\Omega} | F = P(\omega) \text{ for some } \omega\}$, and χ_A is the characteristic function of any set $A \subset \Omega$.

Although positively balanced information structure is weaker than partitional structure, it does not necessarily imply RT-information structure.⁷ Therefore our theorem under RT-information structure is

 $^{^7\}mathrm{See}$ Geanakoplos (1989, p.19) for these relations.

obtained under a different setting in which the information structure is reflexive and transitive but not positively balanced.

In fact, we can construct an economy \mathcal{E} with RT-information structure consisting of a single contingent commodity, $N = \{1, 2\}, \Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}.$

Further, the economy has the property: the RT-information structure is not positively balanced and the endowments are allocated ex ante Pareto-optimally. For the feasible trade $\mathbf{t} = (t_i)_{i \in N}$, $K_C(Acp(\mathbf{t}) \cap R(\mathbf{t})) = \{\omega_4\}$, and zero trade occurs at only the state ω_4 . Note that non-zero trades, however, occur at ω_1, ω_2 , and ω_3 , instead $K_C(Acp(\mathbf{t})) = \Omega$. In addition, we do not specify the traders' attitudes toward risk like Geanakoplos (1989), but unlike several other papers such as Milgrom and Stokey (1982), or Sebenius and Geanakoplos (1983). This means that the crucial character of utility is strict quasi-concavity or monotonicity.

On the whole, what role does the rationality of expectations play in our model? Since, under this concept, each trader knows his expected utility of a given trade, a relationship is stipulated between traders' information structure and expected gains. This approach is similar to the non-partition information technique of Aumann's disagreement theorem.

This paper has examined the no trade theorem under RT-information structure by introducing the concept of rationality of expectations. Although this situation has been investigated by Geanakoplos (1989), our no trade theorem is shown under a slightly different setting as illustrated above, i.e., not positively balanced but RT-information structure.

The Bibliography

- 1. Geanakoplos, J. (1989) "Game theory without partitions, and applications to speculation and consensus" Cowles Foundation Discussion Paper No. 914 (Available on http://cowles.econ.yale.edu).
- 2. Luo, X., and C. Ma (2003) "Agreeing to disagree" type results: a decision-theoretic approach" *Journal of Mathematical Economics* **39**, 849-861.
- 3. Milgrom, P. and N. Stokey (1982) "Information, trade and common knowledge" *Journal of Economic Theory* **26**, 17–27.
- 4. Neeman, Z. (1996) "Common beliefs and the existence of speculative trade" *Games and Economic Behavior* **16**, 77–96.
- 5. Sebenius, J. K. and J. Geanakoplos (1983) "Don't bet on it: Contingent agreements with asymmetric information" *Journal of the American Statistical Association* **78**(382), 424–426.

Variable Selection Criteria in Efficient Decision Trees Construction

Masahiro Miyakawa, Hisayuki Tatsumi

Tsukuba University of Technology, 4-12-7 Kasuga, Tsukuba-shi, Japan 305-0821, {mamiyaka,tatsumi}@cs.k.tsukuba-tech.ac.jp

Nobuyuki Otsu

National Institute of AIST 1-1-1 Umezono, Tsukuba, Ibaraki-ken 305-8568, Japan otsu.n@aist.go.jp

Ivo G. Rosenberg

Université de Montréal, C.P.6128, Succ. "Centre-ville", Montréal P.Q. H3C 3J7, Canada rosenb@DMS. Umontreal.ca

We consider how to convert a given decision table to an efficient decision tree, i.e., to a tree with a small number of nodes. We investigate three variable selection heuristics Γ_A , Γ_H and Γ_D motivated by the three different standpoints of combinatorial, entropy and discriminant analyses, where especially Γ_D is entirely new. The main concern is their behavior, i.e., the rejection of nonessential variables (nev-freeness), the selection of a totally essential variable (tev-boundness) and the selection of a quasi-decisive variable (qdv-boundness). We show that Γ_A is nev-free, tev-bound but not qdv-bound, while Γ_H and Γ_D have the complementary properties. This analysys seems to explain the following experimental performance of the heuristics. An experiment to evaluate the performance of the heuristics shows that each of the heuristics gives near-optimum trees, indicating that Γ_D and Γ_H are practically comparative

and Γ_A is slightly better than the others. All the heuristics require at most $O(L^2 2^L)$ operations with $O(L2^L)$ storage, where L is the number of variables of the input table. Though part of the experimental results were published elsewhere, in this paper we give full proofs for the theoretical part.

Introduction

In many applications we have a finite set O of objects and a list P_1, \ldots, P_L of well-defined logic properties of the objects. The *characteristic vector* of $o \in O$ is $\chi(o) = (b_1, \ldots, b_L)$ where $b_i = 1$ if o satisfies P_i and $b_i = 0$ otherwise. Clearly χ maps O into $\mathbf{2}^L$ where $\mathbf{2} := \{0, 1\}$ and $\mathbf{2}^L$ is the set of zero-one L-vectors.

Next we are given distinct nonempty subsets S_1, \ldots, S_K of $\mathbf{2}^L$ which may called *action classes*, where the indices i of S_i is usually called *actions*. Without loss of generality we may assume that the sets S_1, \ldots, S_K covers $\mathbf{2}^L$ since if they do not, we can add the set $S_{K+1} = \mathbf{2}^L \setminus (S_1 \cup \cdots \cup S_K)$ to S_1, \ldots, S_K . We refer to the cover $\{S_1, \ldots, S_K\}$ of $\mathbf{2}^L$ as a *decision table*. Decision table is just a function F(x) of Boolean variables x with arbitrary values. For our purposes a decision table will be modeled as a sequence (f_1, \ldots, f_K) of Boolean L-ary functions where for all $(a_1, \ldots, a_L) \in \mathbf{2}^L$ we set $f_i(a_1, \ldots, a_L) = 1$ if $(a_1, \ldots, a_L) \in S_i$ and 0 otherwise.

Example 1.. Let L = 2 and $S_1 = \{(0,0), (0,1), (1,1)\}$, $S_2 = \{(0,1)\}$ and $S_3 = \{(0,0), (1,0), (1,1)\}$. The decision table (f_1, f_2, f_3) is on Table 1.

Table 1: An example decision table.

x_1	x_2	f_1	f_2	f_3
0	0	1	0	1
0	1	1	1	0
1	0	0	0	1
1	1	1	0	0

Decision tables frequently occur in many areas; for example, they are used to identify or classify specimens in biology, geology etc. Decision tables are widely used as a conceptual tool, e.g., in pattern recognition, knowledge engineering or in computer programs where they are called Ianov schemes (see [8]).

For a given object o the task is to find the action classes S_i to which $\chi(o)$ belongs. An immediate approach is (i) find $\chi(o)$ by testing the properties one by one and (ii) by inspection find the action classes containing the vector $\chi(o)$. A drawback of this sequential test procedure [15] is that often the decision $\chi(o) \in S_i$ can be made by testing only some of the properties. A natural and often economical representation of a decision table is a decision tree introduced in the next section. We present three heuristics for decision trees in aiming to minimize the number of nodes. The first two are implicit in the literature while the third one is new and motivated by discriminant analysis. The first one is general while the other two only apply when the sets S_1, \ldots, S_K partition $\mathbf{2}^L$. Our main concern is to theoretically analyze how efficient is the variable selection according to the heuristics for essential, totally essential and quasi-decisive variables whereby the last one is a new concept.

Remark. The results presented in this paper were first announced by the first author in the conference Fundamentals of Computation Theory 1987 (Kazan) with no publication. A part of it including the experimental results with the theoretical framework was later presented in [13] but all the proof parts were omitted. In this paper we have kept some of the old experimental part, but our intention is to give the full description of the proofs of the behavior analysis of the algorithms.

Decision trees

Let a decision table F map 2^L into a set B (B is a fixed arbitrary set). A convenient way to represent F is a decision tree. We need the following terminology.

As usual a rooted binary tree T here is a connected acyclic directed graph defined by the set of nodes V and the set of edges E with a specific node r called the root of T. The set V is divided into the two sets: query (or internal) nodes Q and leaves \mathcal{L} where the set Q consists of nodes with two emanating edges and all are of degree 3 with the sole exception of r which has no incoming edge and hence with degree 2 while the set \mathcal{L} consists of nodes of indegree 1 (i.e., leaf is a node with no emanating edges). For

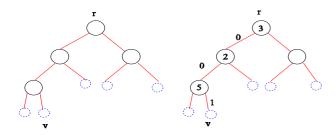


Figure 2: A decision tree

 $v \in V$ denote by π_v the set of vertices on the unique path from r to v and set $\pi_v^* = \pi_v \setminus v$. For $v \in Q$ the two edges emanating from v are labeled by 0 and 1.

There is a map λ from Q into $P = \{1, \ldots, L\}$ taking distinct values on π_v for each $v \in Q$. Denote by σ_v the set of values of λ on π_v and to v assign the following map p_v from σ_v into $\mathbf{2}$. For a node $w \in \pi^*$ set $p_v(\lambda(w)) = i$ if i is the zero-one label of the edge on the path from r to v and emanating from w. The map p_v has the following meaning. First $j = \lambda(w)$ is the index of the variable x_j on which we branch at w and $p_v(j)$ is the value assigned to x_j on the path. An example is on Fig. 1 and the corresponding p_v is on Tab. .

Table 2: A map
$$p_v: \lambda(v) \to \mathbf{2}$$

$$\begin{array}{c|ccccc} x & 3 & 2 & 5 \\ \hline p_v(x) & 0 & 0 & 1 \\ \end{array}$$

Let p_v be a map from a proper subset D of P into $\mathbf{2}$ and $P \setminus D = \{j_1, \ldots, j_k\}$ where $k \geq 1$ and $1 \leq j_1 < \ldots < j_k \leq L$. From $F: \mathbf{2}^L \to B$ we obtain a map $F^{p_v}: \mathbf{2}^k \to B$, called the p_v -residual of F, by fixing certain arguments according to p_v . More precisely, for all $(a_1, \ldots, a_k) \in \mathbf{2}^l$ set

$$F^p(a_1,\ldots,a_k)=F(b_1,\ldots,b_n)$$

where $b_i = p_v(i)$ for all $i \in D$ and $b_{j_l} = a_l$ for all $l = 1 \dots, k$. For example, for L = 5 and p_v from Fig.1 we get for all $(a_1, a_2) \in \mathbf{2}^2$

$$F^{p_v}(a_1, a_2) = F(a_1, 0, 0, a_2, 1).$$

If D = P then F^{p_v} is the element $F(p_v(1), \dots, p_v(L))$ of B.

A node $v \in Q \cup \mathcal{L}$ is steadfast if F^{p_v} is constant, i.e., there exists $b \in B$ such that $F^{p_v}(a_i, \ldots, a_k) = b$ for all $(a_1, \ldots, a_k) \in \mathbf{2}^k$. If p_v maps P into $\mathbf{2}$ then v is obviously steadfast. A steadfast node is a leaf of T to whom we assign the value b. The tree T is a decision tree for F if \mathcal{L} is exactly its set of steadfast nodes. It is easy to see that any $(a_1, \ldots, a_L) \in \mathbf{2}^L$ induces the map $i \mapsto a_i$. This induces a path in T leading to a leaf l of F and $F(a_1, \ldots, a_n)$ is the value assigned to l.

Remark. A binary decision tree T be turned into a binary decision diagram by fusing every pair of nodes v and w with $F^{p_v} = F^{p_w}$.

In general a given decision table possesses many decision trees. Several costs of a decision tree have been proposed; e.g., the average path length [14] or the e-cost [11]. A minimum-cost tree for a given table can be found by dynamic programming techniques requiring $O(L3^L)$ operations (comparisons) with $O(3^L)$ storages [1]. It is known that the classes of various optimum decision tree problems are NP-complete [5] provided the given input table is not in an "expanded" form.

In this paper the *cost* of a tree T is the number |T| of nodes in Q. For $1 \le i \le L$ and $\mathbf{x} = (x_1, \dots, x_L) \in \mathbf{2}^L$ set

$$x^{i} = (x_{1}, \dots, x_{i-1}, \bar{x}_{i}, x_{i+1}, \dots, x_{L})$$

where \bar{a} stands for the negation 1-a of $a \in \mathbf{2}$. The *i*-th variable is essential for F if $F(c) \neq F(c^i)$ for some $c \in \mathbf{2}^L$ and nonessential otherwise. Clearly for F and its residuals only essential variables should be selected. The *i*-th variable is totally essential (t.e.) if for every $c \in \mathbf{2}^L$ we get $F(c) \neq F(c^i)$. It is easy to check that for a t.e.-variable x_i every path from the root r to a steadfast leaf contains a node labeled i and thus it is in our interest to select all t.e.-variables first. Moreover, a t.e.-variable for F remains a t.e.-variable for any of its residuals. However, their detection is not always easy.

The *i*-th variable is a quasi-decisive (q.d.) for F if there is $a \in \mathbf{2}^L$ such that $F(x_1,\ldots,x_{i-1},a,x_{i+1},\ldots,x_L)$ is a constant (from B) while $F(x_1,\ldots,x_{i-1},\bar{a},x_{i+1},\ldots,x_L)$ is not constant. For a q.d.-variable i it is in our interest to set $\lambda(v)=i$ because then the edge emanating from v and labeled a ends in a steadfast leaf.

Call the *i*-th variable optimal if there is an optimal decision for F tree having i at its root. For our cost criterion it is known that both t.e.- and q.d.-variables are optimal and so a minimal tree may always selects a q.d.-variable whenever it appears [11]. A heuristic ideally should shun the nonessential variables and select a t.e.- or a q.d.-variable whenever it arises (it is easy to see that the presence of t.e.- or q.d.-variables among the variables is exclusive; i.e., if there is a q.d.- then no other variable is t.e.-variable and vice versa). In the next section we present 3 variable selection criteria: the activity heuristic Γ_A , the entropy heuristic Γ_H and the discriminant heuristic Γ_D , and behavior analysis for them. Such analysis is important to understand to which degree the heuristics are efficient.

The activity heuristic

This heuristic is implicit in early literature, see e.g., [17,14]. For L > 1 and $i = 1, \ldots, L$ the *i*-th activity is

$$A_i(\mathcal{S}) := \frac{1}{2} |\{ c \in \mathbf{2}^L : \ F(c) \neq F(c^i) \}|$$
 (8)

(recall that $(c_1, \ldots, c_L)^i = (c_1, \ldots, c_{i-1}, \bar{c_i}, c_{i+1}, \ldots, c_L)$; in other words A_i counts the number of pairs (c, c^i) with distinct F(c) and $F(c^i)$.

Clearly $0 \le A_i \le 2^{L-1}$ for all $1 \le i \le L$. It is easy to see that $A_i = 2^{L-1} \Leftrightarrow$ the *i*-th variable is totally essential. Also $A_i = 0$ if and only if and $F(c) = F(c^i)$ for all $c \in \mathbf{2}$ L meaning that i is a nonessential variable for F. Thus both $A_i = 0$ and $A_i = 2^{L-1}$ provide useful information for variable selection.

Lemma 2. Let L > 1. If i is a quasi-decisive variable for F then $A_i = \max(A_1, \ldots, A_L)$. The converse is not true in general.

Proof. Let i be a q.d.-variable. In view of the symmetry of $\mathbf{2}^L$ we can assume i=1. For notational simplicity let $F(0,x_2,\ldots,x_L)$ be constant with value d. Set $X=\{(a_1,\ldots,a_{L-1})\in\mathbf{2}^{L-1}: F(1,a_1,\ldots,a_{L-1})=d\}$ and $Y=\mathbf{2}^{L-1}\setminus X$. Let x=|X| and y=|Y|. First $A_1=y$. Indeed let $c\in\mathbf{2}^L$ satisfy $F(c)\neq F(c^1)$. The notation can be chosen so that $c=(0,c_2,\ldots,c_n)$. Then $d=F(c)\neq F(c^1)$ shows $c^1\in Y$. Conversely, if $c^1\in Y$ then $c=(0,c_2,\ldots,c_n)$ and $d=F(c)\neq F(c^1)$. Due to the symmetry of $\mathbf{2}^L$ it suffices to show:

Claim. $y \geq A_2$.

Proof of the claim: Let $c = (c_1, \ldots, c_L) \in \mathbf{2}^L$ satisfy $F(c) \neq F(c^2)$. Then $c_1 = 1$ since for $c_1 = 0$ we get $F(c) = d = F(c^2)$. By the same token $c \in Y$ or $c^2 \in Y$. Set $Z = \{c \in Y : c^2 \in Y, F(c) \neq F(c^2)\}$, $T = \{c \in Y : c^2 \in X\}$. Clearly, $|Z| \leq y - |T|$ and hence

$$A_2 = \frac{1}{2}|Z| + |T| \le \frac{1}{2}(y + |T|). \tag{9}$$

Suppose to the contrary that $y < \frac{1}{2}(y + |T|)$. Then y < |T|, in contradiction to $T \subseteq Y$. This proves the claim and $A_1 = \max(A_1, \ldots, A_L)$. Finally the Boolean function $F(x_1, \ldots, x_L) = x_1 \oplus \cdots \oplus x_L$ (where \oplus is the EXOR or sum mod 2) satisfies $A_i = 2^{L-1}$ for all $i = 1, \ldots, L$ but has no q.d.-variable.

We calculate A_1, \ldots, A_L . If $A_i = 0$ then the *i*-th variable is nonessential. Set $m = \max(A_1, \ldots, A_L)$. If $m = 2^{L-1}$ then all variables *i* with $A_i = 2^{L-1}$ are totally essential and should be chosen first. Let $m < 2^{L-1}$. We should check whether any of the variable *i* with $A_i = m$ is quasi-decisive. If there is one, we choose it; otherwise we choose the least *i* such that $A_i = m$.

The entropy heuristic

In sections 4 and 5 the function F maps $\mathbf{2}^L$ into $\mathbf{2}^K$ and hence there exist L-ary boolean functions f_1, \ldots, f_K such that for all $x \in \mathbf{2}^L$

$$F(x) = (f_1(x), \dots, f_K(x)).$$

Moreover, we assume that for all $x \in \mathbf{2}^L$

$$\sum_{k=1}^{K} f_k(x) = 1,$$

in other words, the sets $S_k = f_k^{-1}(1)$ partition $\mathbf{2}^L$ into K blocks S_1, \ldots, S_K .

For k = 1, ..., K put $s_k = |S_k| = \sum_{x \in \mathbf{Z}^L} f_k(x)$ and further put $s = s_1 + ... + s_K = 2^L$. The ratio $w_j = s_j/s$ may be seen as the probability of the occurrence of the action j. The corresponding *entropy* is

$$H = -w_1 \log w_1 - \dots - w_K \log w_K \tag{10}$$

(as usual, here and in the sequel $0 \log 0$ is defined to be 0 and the base of the logarithm is 2). By a direct calculation (using $s_1 + \ldots + s_K = s$)

$$H = (s \log s - s_1 \log s_1 - \dots - s_K \log s_K)/s.$$
(11)

The next fact follows from well-known properties of entropy and shows that H is a measure of nondeterminacy or ambiguity.

Fact. (i) $0 \le H \le \log K$, (ii) H = 0 if and only if K = 1 and (iii) $H = \log K$ if and only if $s_1 = \ldots = s_K$.

For $l=1,\ldots,L$ and $b\in\mathbf{2}$ denote by F^{lb} the map obtained from F by fixing the l-th variable to b (i.e., the p_b -residual of F where $p_b:\{l\}\to\{b\}$). Further denote by H_{lb} the entropy of F^{lb} and set

$$H_l = \frac{1}{2}(H_{l0} + H_{l1}). \tag{12}$$

Put $h = \min(H_1, \ldots, H_L)$. The entropy heuristic Γ_H chooses the *i*-th variable where *i* is the least index *k* such that $H_k = h$. It is in [16] and used in e.g., [4,9]. We study certain properties of Γ_H .

From the above fact we obtain:

Lemma 3. Let K > 1 and $H_i = h = \min(H_1, \dots, H_L)$. Then the following are equivalent.

- (i) h = 0
- (ii) $H_j = 0 \Leftrightarrow j = i \text{ for all } 1 \leq j \leq L$,
- (iii) i is the unique essential variable of F.

For $1 \le l \le L$, $1 \le k \le K$ and $b \in \mathbf{2}$ set

$$s_k^{lb} = \sum_{(x_1, \dots, x_{L-1}) \in \mathbf{2}^L} f_k(x_1, \dots, x_{l-1}, b, x_l, \dots, x_{L-1}).$$

Lemma 4. Let $1 \le l \le L$. Then $H_l \le H$ with equality exactly if $s_k^{l0} = s_k^{l1}$ for all k = 1, ..., K. In particular, if l is a nonessential variable, then $H_l = H$ but the converse is not true in general.

Proof. We calculate H_1 . For k = 1, ..., K set

$$p_k = 2^{1-L} s_k^{10}, \quad p_k' = 2^{1-L} s_k^{11}.$$

Notice that

$$p_k + p_k' = 2^{1-L} s_k = 2w_k.$$

Now

$$H_1 = -\frac{1}{2} \left(\sum_{k=1}^{K} p_k \log p_k + \sum_{k=1}^{K} p_k' \log p_k' \right) = -\frac{1}{2} \sum_{k=1}^{K} (p_k \log p_k + (2w_k - p_k) \log(2w_k - p_k)).$$
 (13)

For 0 < a < 1 consider the function

$$f_a(x) = -x \log x - (2a - x) \log(2a - x)$$

defined on [0,2a]. Clearly $f_a(x)=f_a(2a-x)$ on [0,2a]. Its second derivative is $f_a''(x)=-2(\ln 2)^{-2}ax^{-1}(2a-x)^{-1}$ which is negative on (0,2a). It follows that f_a has a unique maximum at x=a and this maximum value is $f_a(a)=-2a\log a$. Applying this to (13) we see that

$$H_1 \le -\sum_{k=1}^K w_k \log w_k = H.$$

For equality clearly we need $p_k = w_k$ for all k = 1, ..., K. This means

$$s_k^{10} = s_k^{11} \quad (k = 1, \dots, K) \tag{14}$$

Clearly (14) holds whenever the first variable of F is nonessential. Example 2 shows that the converse is not true.

Lemma 5. Let F have a quasi-decisive variable. Then for $1 \le l \le L$ we have $H_l = h = \min(H_1, \dots, H_L)$ if and only if l is a quasi-decisive variable of F.

Proof. For notational simplicity let the first variable of F be quasi-decisive and let F^{10} be constant. Then the value of F^{10} is the element $(1,0,\ldots,0)$ of $\mathbf{2}^K$ because the sets S_1,\ldots,S_K partition $\mathbf{2}^K$). It follows that $H_{10}=0$. Let $p=2^{L-1}$. The map F^{11} corresponds to the sets $S_1\setminus\mathbf{2}^{L-1},S_2,\ldots,S_K$ where $S_1\setminus\mathbf{2}^{L-1}$ may be empty. Now from (11)

$$F^{11} = \{ (S_1^{11}, i_1), (S_2, i_2), \dots, (S_K, i_K) \}$$

where S_1^{11} may be empty. Put $s_l = |S_l|$ for l = 1, ..., K and note that $|S_1^{11}| = s_1 - m$. We have $H^{i0} = 0$ and from

$$2H_1 = 2H_{11} = \log p - ((s_1 - p)\log(s_1 - p) + \sum_{j=2}^{K} s_j \log s_j)/p.$$
(15)

Let F^{20} and F^{21} correspond to the covers of $\mathbf{2}^{L-1}$ by pairwise disjoint sets M_1, \ldots, M_K and N_1, \ldots, N_K , respectively. For $k = 1, \ldots, K$ put $m_k = |M_k|$ and $n_k = |N_k|$. Note that

$$m_k + n_k = s_k \quad (k = 1, \dots, K).$$
 (16)

From the definition and (11) we have

$$2H_2 = (2p\log p - \sum_{l=1}^{K} (m_l \log m_l + n_l \log n_l))/p.$$
(17)

To show that $H_1 \leq H_2$ we prove that

$$\sigma = 2p(H_2 - H_1) \ge 0.$$

From (15)

$$\sigma = p \log p + (s_1 - p) \log(s_1 - p) - m_1 \log m_1 - n_1 \log n_1 + \sum_{j=2}^{K} (s_j \log s_j - m_j \log m_j - n_j \log n_j).$$
(18)

We show that $s_j \log s_j - m_j \log m_j - n_j \log n_j$ is always non-negative. If $m_j = 0 \neq n_j$ then this follows from the fact that $s_j \geq n_j$ and the real function $g(x) = x \log x$ is increasing on $[1, \infty)$. The same argument applies if $m_j \neq 0 = n_j$. Thus assume that $m_j \neq 0 \neq n_j$. Taking into account (16) and $s_j/m_j \geq 1, s_j/n_j \geq 1$ for all $j = 2, \ldots, K$,

$$s_i \log s_i - m_i \log m_i - n_i \log n_i = m_i \log(s_i/m_i) + n_i \log(s_i/n_i) \ge 0.$$
 (19)

We show that the first part of (18) is also nonnegative. For this define a real function t on $[0, s_1]$ by setting for all $0 \le x \le s_1$

$$t(x) = (s_1 - x)\log(s_1 - x) + x\log x.$$

Clearly $t(x) = t(s_1 - x)$ and t is increasing on $[\frac{1}{2}s_1, s_1]$. Put $a = \max(m_1, n_1)$. From (16) we have $m_1 + n_1 = s_1$ and so the first part of (18) can be written as t(p) - t(a). By (16) we have

$$s_1 \ge p \ge a \ge \frac{1}{2}(m_1 + n_1) = \frac{1}{2}s_1$$

This concludes the proof of $\sigma \geq 0$ and $H_1 \leq H_2$. By the symmetries of $\mathbf{2}^L$ this shows $H_1 = h$. Suppose to the contrary that $H_1 = H_2$. We have shown that the right side of (18) consists of nonnegative terms. Then, in particular, t(p) = t(a). Since $\frac{1}{2}s_1 \leq a \leq m \leq s_1$ and t is strictly increasing on $[\frac{1}{2}s_1, s_1]$, we have a = p. This, in fact, means that 2 is a quasi-decisive variable.

The discriminant heuristic Γ_D

In a decision table each variable x_i contributes to discriminate between different actions. Hence one can measure "discriminating power" of a variable x_i from the standpoint of the discriminant analysis [3]. We introduce a heuristic based on this idea which, to the authors' knowledge, is new. First we define the following parameter. Let F be a map from $\mathbf{2}^L$ into $\mathbf{2}^K$ such that for all $k=1,\ldots,K$ the numbers $s_k=|S_k|>0$ and

$$s_1 + \ldots + s_K = 2^L.$$
 (20)

Let $1 \le l \le L$ and let F^{l_1} correspond to the sets T_1, \ldots, T_K . Note that $t_k = |T_k|$ satisfy

$$0 \le t_k \le s_k \quad (j = 1, \dots, K), \quad t_1 + \dots + t_K = 2^{L-1}.$$
 (21)

Denote the "interclass" variance of the l-th variable by setting

$$\sigma_l = 2^{-L} \sum_{k=1}^K (t_k - \frac{1}{2}s_k)^2 / s_k.$$
 (22)

Using (20) a straight-forward calculation shows that

$$\sigma_l = -1/4 + 2^{-L} \sum_{k=1}^{K} t_k^2 / s_k. \tag{23}$$

We determine sharp bounds for σ_l .

Lemma 6. (i) We have $0 \le \sigma_l \le 1/4$ for all $1 \le l \le L$,

- (ii) $\sigma_l = 0$ if and only if $t_k = \frac{1}{2}s_k$ for all $k = 1, \dots, K$, (iii) $\sigma_l = 1/4$ if and only if $t_k \in \{0, s_k\}$ for all $k = 1, \dots, K$.

Proof. For $k=1,\ldots,K$ put $x_k=t_k-\frac{1}{2}s_k$. Then (21)–(22) become

$$|x_k| \leq \frac{1}{2} s_k \quad (k = 1, \dots, K), \tag{24}$$

$$x_1 + \dots + x_K = 0, \tag{25}$$

$$x_1 + \dots + x_K = 0,$$
 (25)
 $\sigma_l = 2^{-L}(x_1^2/s_1 + \dots + x_K^2/s_K).$ (26)

From (24) we obtain $x_k^2/s_k = |x_k|^2/s_k \le \frac{1}{4}s_k$ and from (26) we get

$$\sigma_l \le 2^{-L} (s_1 + \ldots + s_K)/4 = \frac{1}{4}.$$

Now (i) follows from this and (26). Next by (24) and (26) we have $\sigma_l = \frac{1}{4}$ exactly if $|x_k| = \frac{1}{2}s_k$ for all $k=1,\ldots,K$ which means $t_k-\frac{1}{2}s_k=\pm\frac{1}{2}s_k$ and $t_k\in\{0,s_k\}$. Combining this with (25) we obtain (iii). Finally (ii) follows from (26).

Now we define the discriminant heuristic Γ_D . At each step it chooses the first variable i such that

$$\sigma_i = \max(\sigma_1, \dots, \sigma_L). \tag{27}$$

We investigate the behavior of Γ_D with respect to nonessential, totally essential and quasi-decisive vari-

Lemma 7. If the l-th variable is non-essential then $\sigma_l = 0$. In general $\sigma_1 = 0$ may hold even for a totally essential variable.

Proof. Assume l-th variable to be nonessential. Then we have $t_k = \frac{1}{2}s_k$ for each k and hence $\sigma_l = 0$ from Lemma 6(ii). For the second see Example 2

Example 2.. Let L = 4, K = 3 and let F be given by

$$\begin{array}{lll} S_1 & = & \{(0,0,0,0),(0,0,1,1),(0,1,1,0),(1,0,0,1),(1,1,0,0),(1,1,1,1)\}, \\ S_2 & = & \{(0,0,0,1),(0,0,1,0),(0,1,1,1),(1,0,0,0),(1,1,0,1),(1,1,1,0)\}, \\ S_3 & = & \{(1,0,1,0),(1,0,1,1),(0,1,0,0),(0,1,0,1)\}. \end{array}$$

To calculate H_1 we exhibit on Tab. 2 the tables for F^{10} and F^{11} . On Tab. 2 in the last two columns we write 1,2,3 instead of (1,0,0),(0,1,0),(0,0,1), respectively.

The variable 1 is not totally essential due to the 3rd line or the 6th line. Now $H_{10}=H_{11}=\frac{3}{8}\log\frac{3}{8}+\frac{3}{8}\log\frac{3}{8}+\frac{2}{8}\log\frac{2}{8}\doteq1.56$ and so $H_1=\frac{1}{2}(H_{10}+H_{11})\doteq1.56$. Next $s_j^{10}=s_j^{11}=\frac{1}{2}s_j$ for k=1,2,3 leads to $\sigma_1=0$. The status of the variables and the values needed for three heuristics are on Tab. 3. Note that the variables 2,3,4 are totally essential.

Lemma 8. If the *i*-th variable of F is quasi-decisive then for every $1 \le k \le K$

$$\sigma_k = \max(\sigma_1, \dots, \sigma_L)$$

if and only if k is a quasi-decisive variable.

x_2	x_3	x_4	F^{10}	F^{11}
0	0	0	1	2
0	0	1	2	1
0	1	0	3	3
0	1	1	1	2
1	0	0	2	1
1	0	1	3	3
1	1	0	1	2
1	1	1	2	1

variable	essentiality	A_i	H_i	σ_i
1	non-tev	6	1.56	0
2	tev	8	1.56	0
3	tev	8	1.56	0
4	tev	8	1.56	0

Proof. For notational simplicity let the first variable be quasi-decisive and let F^{10} be constant. As in the proof of Lemma 5 we obtain that the constant value of F^{10} is 1. We use the same notation as in the proof of Lemmas 5 and 6. Put

$$\nu = s_1 \cdot \ldots \cdot s_K, \ s'_k = \nu / s_k \ (k = 1, \ldots, K), \ \lambda = n_2 + \ldots + n_K.$$
 (28)

We show that

$$h_k = 2^L \nu(\sigma_1 - \sigma_2) \ge 0.$$

First from $S_1 \supset \{0\} \times \mathbf{2}^{L-1}, S_2 \cup \cdots \cup S_K \subseteq \{1\} \times \mathbf{2}^{L-1}$ and (23) we get

$$\sigma_1 = -\frac{1}{4} + 2^{-L} \left[\frac{(s_1 - m)^2}{s_1} + \sum s_k \right]$$

(here and in the sequel all sums are over k = 2, ..., K)

From (23)

$$\sigma_2 = -\frac{1}{4} + 2^{-L} \left[\frac{n_1^2}{s_1} + \sum \frac{n_k^2}{s_k} \right]$$

and thus

$$2^{L}(\sigma_{1} - \sigma_{2}) = s_{1} - 2m + \frac{m^{2} - n_{1}^{2}}{s_{1}} + \sum s_{k} - \sum \frac{n_{k}^{2}}{s_{k}}.$$
 (29)

Using $s_1 + \ldots + s_K = 2m, n_1 = m - \lambda$ and multiplying (29) by ν

$$\begin{array}{lcl} h & = & s_1'\lambda(2m-\lambda) - \sum s_k'n_k^2 \\ & = & s_1'\lambda(2m-\lambda) - \nu\lambda + \nu\lambda - \sum s_k'n_k^2 \end{array}$$

From (28) we get $\nu\lambda = \sum s_k' s_k n_k$ and $\nu = s_1' s_1$ and hence

$$h = s'_1 \lambda (2m - \lambda - s_1) + \sum s'_k n_k (s_k - n_k).$$

Using $s_1 + ... + s_K = 2m$ and (28)

$$h = s_1' \lambda (2m - \lambda - s_1) + \sum s_k' n_k (s_k - n_k).$$
 (30)

This is a sum of non-negative terms and so $h \ge 0$. The equality h = 0 holds if and only if $s_k = n_k$ for each k = 2, ..., K. This means that the second variable k is also quasi-decisive.

Experimental results

We have compared the performance of the three heuristics Γ_A , Γ_H and Γ_D on the decision tables for which the occurrence of actions are setup in the following way (so called random 80-20 distributions). In most practice, occurrence frequencies of actions in a table are far from uniform. So we assumed the well-known 80-20 rule [6] which states that 80 percent of occurrences of the actions is concentrated to the

Heuristic	DP		Γ_A		$\Gamma_H, \ \Gamma_D$		random	
Average costs	282.0	(11.5)	289.5	(12.2)	296.2	(13.8)	362.3	(12.1)

Criteria	random	Γ_D	Γ_H	Γ_A	DP
Average time (second)	1.02	2.01	2.03	2.59	16.60

most active 20 percent of actions; and the same rule applies to this 20 percent recursively. Each action is distributed randomly in the initial table (2^L vectors). We generated them for L=4 through 10 (the number of variables) and K=2 through 20 (the number of actions) for the experiment.

A. Performance comparison

The average is taken over the costs of the constructed trees over 1000 input random 8-2 tables for each heuristic and of the costs of optimum trees (obtained by the DP algorithm) over 100 such tables. It shows that the performance of Γ_H and Γ_D is practically identical and Γ_A is better than both of them (for the $L=10,\ K=4$ case, 1.03 versus 1.05 in terms of "optimality" coefficient, which is defined by the ratio of the average cost of VSM-trees to the average cost of optimal trees). The result of the experiments is indicated in Fig. 2. The average costs of the trees for the case of $K=4,\ L=10$ is given in the following table (standard deviations in the parentheses).

In the following table we also give an average time (over 1000 trees) required for constructing a tree for each method for the case of K=4, L=10. The experiment was performed on a FACOM M200 computer executing about 7 MIPS (no code-optimization was attempted).

B. Implementation and the amount of computation

We assume that the table's action classes S_j partition $\mathbf{2}^L$. We compute A_i , H_i and σ_i for each variable i. By (8),(12) and (22) all of them depend on its subtables. Assume that all 2^L actions are stored in the memory. To obtain subtables we need to scan every rule of F. So we need computation at least proportional to 2^L . Below we check that no bigger order of computation is needed to compute each of A_i , H_i and σ_i .

- 1. Computation of A_i . We may assume that the vectors of the two subtables F^{i0} and F^{i1} are sorted in the same order (sorting time may be ignored since we can use an implementation generating vectors of F^{i0} and F^{i1} in the same order in time proportional to 2^L). Then we simply compare each pair of actions of the corresponding vectors and count the number of times when they have different action names. Thus 2^{L-1} comparisons are sufficient.
- 2. Computation of σ_i and H_i . First for σ_i . We need counting only of $|T_j|$ in F^{i1} (for $|S_j|$ in F it is known as the process of expanding a tree goes top-down). Next for H_i . We need counting $|S_j|$ in F^{i1} . The $|S_j|$ in F^{i0} can be obtained by subtracting $|S_j|$ in F^{i1} from $|S_j|$ in F (notice the occurrence of actions is preserved in the dividing of table).

So in both H_i and σ_i cases we need exactly the same 2^{L-1} amount of computation.

At each level i of the tree there are at most 2^i tables with 2^{L-i} rules and for each table we need L-i (there are so many number of variables) computing of the criterion. Summing up for $i=0,\ldots,L-1$ (there are at most L levels) the amount of computation required for the algorithm is bounded from above by $O(L^22^L)$. We need $O(L2^L)$ storage if we keep every subtable at each node (cf. [12]).

In the implementation of H_i and σ_i we counted the occurrence of $|S_j|$ in both subtables F^{i0} and F^{i1} . In the implementation of Γ_A we stored essential j-pairs for each variable j in the memory which required extra storage.

Conclusions

We summarize the results.

Theorem 1. (i) Only the heuristic Γ_A rejects non-essential variables and selects a totally essential variable while Γ_D but Γ_H may select a non-essential variable.

(ii) The heuristics Γ_D and Γ_H select a quasi-decisive variable while Γ_A need not select it.

The next example indicates that Γ_H and Γ_D do not give even unique values to totally essential variables, indicating less relevance of these heuristics to total essentiality.

x_1	x_2	x_3	F^{40}	F^{41}
0	0	0	1	3
0	0	1	2	3
0	1	0	2	1
0	1	1	1	2
1	0	0	2	1
1	0	1	1	2
1	1	0	1	3
1	1	1	2	3

variables	essential.	A_i	H_i	σ_i
1	tev	8	1.56	0
2	tev	8	1.56	0
3	non-tev	6	1.5	1/48
4	tev	8	1.25	3/32

Example 3.. We present F by exhibiting its two subtables F^{40} and F^{41} .

We have the following values of the criteria for each variable.

In fact we have $H_2 > H_3 > H_4$ and $\sigma_{B_2} < \sigma_{B_3} < \sigma_{B_4}$ and the variable 3 is not totally essential while the variables 1, 2 and 4 are totally essential. In this example the criteria Γ_D and Γ_H select variable 4 (one of the optimal variables), while Γ_A may select any of 1, 2 and 4.

A. Assessment of the behaviors of the heuristics

As Γ_A shows the best optimality ratio, in order to obtain an efficient tree, the crucial behavior of the heuristic seems to reject the worst variable rather than to select one of the best variables (when among the totally essential or quasi decisive variable). This coincides with the observation given in [12].

B. Relevance to the minimal path-length optimality

In [12] three other VSM-heuristics are given for minimal path-length optimality for a general decision table where in addition variable i carries a positive real testing cost (time) C_i and to each rule an execution probability are attached. (The criteria are derived from a detailed analysis; in particular the criterion Q becomes tev-bound and reduces to Γ_A when C_i is unit (= 1) and each rule is executed with a uniform probability (= 2^{-L}).) Let us distinguish two optimalities by naming time or t-optimality and space or s-optimality. A natural question is the relevance between the two optimalities: how much t-optimality is attained for an s-optimal tree and vice versa. An experiment is done assuming the following two probability distributions for the rules: 1. uniform, 2. 80-20 rule (so in the latter case to each rule of the random s-2 table an execution probability taken randomly from the s-20 distribution is attached). Denote by T_{s -opt (T_{t} -opt) the set of s-optimal (s-optimal) trees and the average of their s-cost (s-cost) by average-scost (average-tcost). (These optimum trees were obtained by DP algorithm). For our purpose the "goodness" of s-optimization and s-optimization may be measured respectively by

$$c_s = \frac{\text{average-tcost}(\mathcal{T}_{\text{t-opt}})}{\text{average-tcost}(\mathcal{T}_{\text{s-opt}})} \text{ and } c_t = \frac{\text{average-scost}(\mathcal{T}_{\text{s-opt}})}{\text{average-scost}(\mathcal{T}_{\text{t-opt}})},$$

where bigger coefficient ≤ 1.0 indicates better performance. We have the following results[10]:

- 1. For the uniform distribution the average is taken over 500 initial tables. Both c_s and c_t are very close to 1.0, e.g., in the N=10, L=4 case $c_s=0.99$ and $c_t=0.98$. So the two optimalities are related very closely.
- 2. For the 80-20 probability distribution average is taken over 500 initial tables (first a single random 8-2 table is generated, then execution probability assignment was done 500 times to its rules resulting in 500 initial tables.) The coefficient c_t is always greater than c_s , e.g., for the N=10, L=4 case, we have $c_t=0.93$ while $c_s=0.89$.

The above observation seems to suggest that in practical situations where one wishes an efficient tree in both optimalities, path-length optimization rather than node-number optimization is more adequate. In this sense the heuristic O in [12] seems to be a best choice (however, no direct experimental comparison of the 6 criteria is done and the comparison may merit another study).

Acknowledgment This work was initiated while the first author was at the Electrotechnical Laboratory. It is supported in part by the Ministry of Education, Science, Sport and Culture, Grant-in-Aid for Scientific Research No.18500700 in the final stage.

The Bibliography

- 1. A.J. Bayes: "A dynamic programming algorithm to optimise decision table code," *Australian Computer J.* 5, 2, 77-79, May 1973,
- 2. R.E. Bryant: "Graph-based algorithm for Boolean function manipulations," IEEE Trans. Comput., C-35, 8, 677-691, Aug. 1986.
- 3. R.A. Fisher: "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, 7, Part II, 179-188, 1936.
- 4. C.R.P. Hartman, P.K. Varshney, K.G. Mehrotra, C.L. Gerberich: "Application of information theory to the construction of efficient decision trees," *IEEE Trans. on Information Theory* IT-28,4, 565-577, July 1982.
- 5. L. Hyafil, R.L. Rivest: "Constructing optimal binary decision trees is NP-complete," *Comm. ACM* 5, 1, 15-17, May 1976.
- D.E. Knuth: The art of computer programming, vol. 1, 2nd ed. Addison-Wesley, Reading, Mass., 1973.
- 7. D.E. Knuth: *The art of computer programming*, vol. 3, Second Edition, Addison-Wesley, Reading, Mass., 1998.
- 8. M. Miyakawa, V.K. Sabelfeld: "On minimizations of size of logical schemes (in Russian)," *Theoretical basis of compiling* (A. P. Ershov, ed.), 49-58, Novosibirsk State University, Novosibirsk 1980.
- 9. M. Miyakawa, N. Otsu: "Algorithms for constructing near-minimum total nodes decision trees from expanded decision tables (Japanese)," *TGEC IECE Japan*, EC82-33, July 1982.
- 10. M. Miyakawa, T. Morioka: On efficiency of decision tree optimization (in Japanese). 1983 Nat. Conv. Rec. IECE Japan 1335, 1983.
- 11. M. Miyakawa: "Optimum decision trees an optimal variable theorem and its related applications -," *Acta Informatica* 22, 475-498, 1985.
- 12. M. Miyakawa: "Criteria for selecting a variable in the construction of efficient decision trees," *IEEE Trans. Comput.*, **38**(1), 130-141, 1989.
- 13. Miyakawa, M., Otsu, N., Rosenberg I.G.: "Variable selection heuristics and optimum decision trees An experimental study -," *Proc. The 32nd International Symposium on Multiple-Valued Logic*, Boston, 2002, 238-244.
 - 14. B.M.E. Moret: "Decision trees and diagrams," Computing Surveys 14, 4, 593-623, Dec. 1982.
- 15. L.T. Reinwald, R.M. Soland: "Conversion of limited-entry decision tables to optimal computer programs I: minimum average processing time," *JACM* 13, 3, 339-358, July 1966.
- 16. L.T. Reinwald, R.M. Soland: "Conversion of limited-entry decision tables to optimal computer programs II: minimum storage requirement," *JACM* 14, 4, 742-755, Oct. 1967.
- 17. K. Schwayder: "Conversion of limited-entry decision tables to computer programs a proposed modification of Pollaks algorithm," *Comm. ACM* **14** (2), 69-73, Feb. 1971.
 - 18. V. G. Sprague: "On storage space of decision trees," Comm. ACM 9 (5), 319-319, May 1966.
- 19. S. Yanuskevich, D. Smovic: "Application of information theory to logic design: results, trends and non-solved problems", The 8th International Workshop on Post-Binary Ultra-Large-Scale Integration Systems (Extended abstracts), Freiburg, May 1999, 25-29.

Agent-based Non-line-of-sight Geolocation of Emitters

Mikhalev A.

A. Department of Aerospace, Power and Sensors, Cranfield University, The Defence Academy of the UK, Shrivenham, Swindon, SN6 8LA, UK a.mikhalev@cranfield.ac.uk

This paper describes a novel agent-based, multi-platform, multi-sensor, emitter geolocation algorithm for emitters that may be non-line-sight or masked by the terrain. Sensors and processing algorithms mounted on UAV, ground vehicles or handheld devices can be treated as independent agents. They may operate independently or cooperatively to provide a range of sensor data such as angle-of-arrival (AOA), time difference of arrival (TDOA), frequency difference of arrival (FDOA) etc. In addition, the sensor data can be fused with terrain data to aid accurate emitter geolocation.

The method of sensor data fusion that has been adopted here is based on the concept of generating the emitter position likelihood for the various types of sensor data. Because the data has been transformed into likelihood functions and is now in a unified parameterised space it is possible to fuse TDOA, AoA and FDOA data to provide improved estimate of the emitter position using the Hough Transform. This method also simplifies the fusion of terrain data, as this may also be specified in terms of the likelihood that the emitter may be located within the search space. In a similar manner any other *a priori* knowledge concerning the emitter can also be fused with the sensor data the stage. Hough Transform parametrised space can be used as a representation of environment for agents or as operational space.

Our research has shown that the results of the proposed method are comparable with high-computationally intensive methods such as the particle filter in terms of rms position error but with lower computational effort. Figure 1 shows the 3D view of the Hough transform space that fuses two TDOA and one AOA measurements. The single peak locates the target emitter.

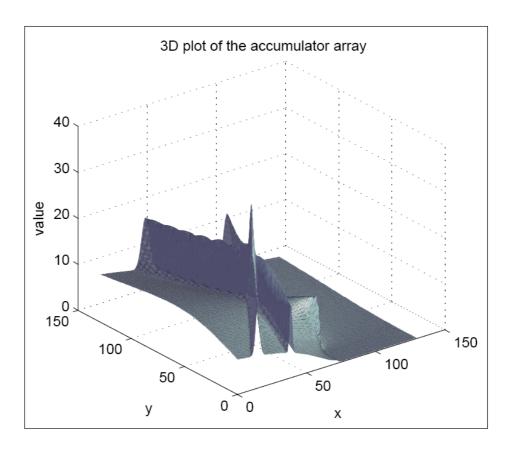


Figure 1 Hough transform of TDOA and AOA data

Study of Clones Irreducible by Means of Lattice Operations

Grant R. Pogosyan¹

Mathematics and Computer Science, International Christian University, Tokyo, Japan e-mail: grant@icu.ac.jp

The Lattice of Clones, i.e. algebraically closed sets of Multiple Valued Logic functions containing all projections, remains mostly unknown. After the pioneering work of E.Post [15] who completely described the lattice in the Boolean case, much effort was directed to achieving a similar result for higher arity logics. Particularly, the first important task was to prove a Post-like completeness criterion by finding all co-atoms of the lattice. For the case of ternary logic this problem was solved by S.V.Yablonskii [19] followed by a remarkable completion of the task in the general case by I.G.Rosenberg [16]. Regarding atoms of the lattice, they are known only for up to the ternary case, for which the task was completed by B.Čsakany [2] using the power of computer calculation.

Due to the overwhelmingly complex nature of the lattice, a comprehensive description of all clones in the general case will be very difficult to achieve in the near future. Meanwhile, further study of this lattice is gaining more importance due to the growing number its applications in different fields of Universal Algebra and Computer Science. A noteworthy relation was recently established between Clone Theory and Computational Complexity Theory. In recent years, much effort has been directed towards understanding and description of certain elements and parts of the lattice, such as chains and monoidal intervals.

In this presentation we focus on the elements of the clone lattice that are totally irreducible by means of the elementary lattice operations – meet and join. Such clones possess certain qualities that trigger interest towards their study. Particularly, irreducible elements can serve as generators of the entire lattice by means of joins and/or meet expressions. The study of irreducible clones began over a decade ago when the author and his colleagues posed certain problems and presented first results [9,10]. Since then some of the questions raised have been answered. Particularly, there were obtained general criteria for both join- and meet-irreducible clones [12] which provide full descriptions of such chains. More results have been found for the monoidal part of the lattice [10] and regarding the generation the Post's lattice [11]. Meanwhile, there remain several important open questions, such as the existence of so-called transit clones, i.e. those irreducible by both meet and join operations.

Here we review the study of irreducible clones presenting the major results obtained so far as well as problems which still remain to be solved.

Definitions and Preliminaries

In the following subsections we present the notations, notions and some basic facts about irreducible elements in general lattices and related algebras, the Clone Lattice of Multiple-Valued Logic functions and its Post version for the Boolean case.

Irreducible Elements in Lattices

Let $\mathcal{L} = \langle L; \wedge, \vee \rangle$ be a lattice whose order relation is denoted by \leq . An element $a \in L$ is join-irreducible if $a = x \vee y$ implies a = x or a = y. Clearly, the least element of \mathcal{L} and the atoms of \mathcal{L} , if they exist, are trivially join-irreducible.

Call $a \in L$ a totally join-irreducible element, shortly a tji-element, of \mathcal{L} if either a is the least element of \mathcal{L} or the set $\{x \in L : x < a\}$ has a greatest element (which will be denoted by \bar{a}).

A meet-irreducible element and a totally meet-irreducible one, shortly a tmi-element, are defined dually; in other words, they are accordingly a join-irreducible and a tji-element of the dual lattice $\mathcal{L}^d = \langle L; \vee, \wedge \rangle$.

Let $\mathcal{A} = \langle A; F \rangle$ be a (non-indexed) algebra, i.e., A is a nonvoid set and F is a set of finitary operations on \mathcal{A} which are maps from A^n into A of arity n with n a nonnegative integer (whereby for n = 0 the nullary operations are identified with the elements of A).

A subset S of A is a *subuniverse* of A if S contains all the nullary operations from F and $f(S^n) \subseteq S$ for every $f \in F$ of a positive arity n. The set Sub(A) of all subuniverses of A, ordered by containment, forms an algebraic lattice Sub(A) in which the meet of any system of subuniverses of A is their set-theoretical intersection

For $X \subseteq A$, the least subuniverse of \mathcal{A} generated by X and is denoted by [X]. For $a \in A$ we abbreviate $[\{a\}]$ by [a]. The join of $\{X_i : i \in I\} \subseteq Sub(\mathcal{A})$ is $[\bigcup_{i \in I} X_i]$.

¹During the period of April 1 - October 31, 2006 the author held a position of Kolmogorov Guest-Professor in Mathematics, Moscow State University, Moscow, Russia

Next, $X \in Sub(\mathcal{A})$ is single generated if X = [x] for some $x \in X$, whereby x is called a single generator of X.

Clones and Mal'tsev Algebras

Let us fix the universe to the set $\mathbf{k} = \{0, 1, \dots, k-1\}$, though generally the definitions below apply to any nonvoid set A. For a positive integer n an n-ary operation on \mathbf{k} or a k-valued logic function is a map from \mathbf{k}^n into \mathbf{k} . An example is the i-th n-ary projection $e_{k,i}^n$ defined by $e_{k,i}^n(a_1, \dots, a_n) = a_i$ for all $a_1, \dots, a_n \in \mathbf{k}$ $(1 \le i \le n)$. Denote by $O_k^{(n)}$ the set of all n-ary operations on \mathbf{k} and set $O_k = \bigcup_{n \ge 1} O_k^{(n)}$. For for formal reasons the nullary operations on \mathbf{k} are replaced by constant unary operations on \mathbf{k} . Set $Q_k = \{e_{k,i}^n : 1 \le i \le n\}$.

Informally, a clone on **k** is a subset of O_k containing Q_k and closed under composition. To specify the notion of composition we use the following elegant and more algebraic definition essentially due to Mal'tsev [8]. It is based on one binary operation * and four unary operations $\zeta, \tau, \Delta, \nabla$ on O_k .

Mal'tsev [8]. It is based on one binary operation * and four unary operations $\zeta, \tau, \Delta, \nabla$ on O_k . Let $f \in O_k^{(m)}$ and $g \in O_k^{(n)}$. The product $h = f * g \in O_k^{(m+n-1)}$ is obtained by replacing the first variable of f by g while keeping all variables of h distinct. For all $a_1, \ldots, a_{m+n-1} \in \mathbf{k}$ set

$$h(a_1, \dots, a_{m+n-1}) = f(g(a_1, \dots, a_n), a_{n+1}, \dots, a_{m+n-1}).$$

The unary operations ζ and τ perform two special variable exchanges:

If m=1 set $\zeta(f)=\tau(f)=f$. For m>1 define $\zeta(f),\tau(f)\in O_k^{(m)}$ by setting for all $a_1,\ldots,a_m\in\mathbf{k}$

$$(\zeta f)(a_1,\ldots,a_m) := f(a_2,\ldots,a_m,a_1),$$

$$(\tau f)(a_1,\ldots,a_m) := f(a_2,a_1,a_3,\ldots,a_m).$$

The unary operation Δ identifies the first two variables. For m=1 we set $\Delta f=f$, while for m>1 define $\Delta f\in O_k^{(m-1)}$ by setting for all $a_1,\ldots,a_{(m-1)}\in\mathbf{k}$

$$(\Delta f)(a_1,\ldots,a_{m-1}) := f(a_1,a_1,a_3,\ldots,a_{m-1}).$$

Finally, ∇ adds a fictitious (or dummy) variable. $\nabla f \in O_k^{(m+1)}$ is defined by setting for all $a_1, \ldots, a_{m+1} \in \mathbf{k}$

$$(\nabla f)(a_1,\ldots,a_{m-1}) := f(a_1,\ldots,a_{m-1},a_m).$$

Call $\mathcal{M}'_k = \langle O_k; *, \zeta, \tau, \Delta, \nabla \rangle$ the *iterative algebra* on **k** [8] and its subuniverses the *closed classes* on **k**. Similarly, one could call $\mathcal{M}_k = \langle O_k; *, \zeta, \tau, \Delta, \nabla, e^1_{k,1} \rangle$ the *Post-iterative algebra* on **k** and its subuniverses the *clones* on **k**. Usually clones are defined as the subuniverses of $\langle O_A; *, \tau, \Delta, e^2_{A,1} \rangle$ but for our purposes \mathcal{M}' is better suited.

Denote by L_k and \mathcal{L}_k the set $Sub(\mathcal{M}'_k)$ and the lattice $\mathbf{Sub}(\mathcal{M}'_k)$ correspondingly.

We are interested in the sets \mathbf{J}_k and \mathbf{M}_k of the totally join- and meet-irreducible elements of \mathcal{L}_k .

Relational Clones and Krasner Algebras

As we mentioned earlier the meet-irreducible and totally meet-irreducible elements of the lattice \mathcal{L} are defined dually as the join-irreducible and totally join-irreducible elements of the dual lattice \mathcal{L}^d . For the lattice \mathcal{L}_k its dual has the representation as the subuniverse lattice of an algebra on the set R_k of the finitary relations on \mathbf{k} [1,3]. Below we provide a brief explanation of this representation.

For a positive integer h a subset of \mathbf{k}^h , i.e. a set of h-tuples over \mathbf{k} is an h-ary relation on \mathbf{k} . Denote by $R_k^{(h)}$ the set of all h-ary relations on \mathbf{k} and set $R_k = \bigcup_{h \geq 1} R_k^{(h)}$. We say that $f \in O_k^{(n)}$ preserves $\rho \in R_k^{(h)}$ if ρ is a subuniverse of A; more explicitly, if for every $h \times n$ matrix $M = [m_{ij}]$ whose columns are all in ρ , the values of f on those rows of M form an h-tuple from ρ . Formally, $(m_{1j}, \ldots, m_{hj}) \in \rho$ implies

$$(f(m_{11},\ldots,m_{1n}),\ldots,f(m_{h1},\ldots,m_{hn})) \in \rho$$

The concept goes back to [4], is known under many names and captures e.g. congruences and endomorphisms. Now "f preserves ρ " defines a binary subrelation of $O_k \times R_k$ and hence induces a Galois connection (or polarity).

First we describe the Galois closed subsets of O_k . For $\rho \in R_k$ set

$$Pol \rho = \{ f \in O_k : f \text{ preserves } \rho \}.$$

It is well known that $\operatorname{Pol}\rho$ is a clone on **k**. We call a clone of this form *rational*. The Galois closed subsets of O_k are exactly the intersections of rational clones, in fact, they coincide with \mathcal{L}_k . This can be

made more precise: a clone C on \mathbf{k} is irrational (i.e., non-rational) exactly if there exist rational clones $C_1 \supset C_2 \supset \cdots$ on \mathbf{k} whose intersection is C.

The Galois closed subsets of $R_{\mathbf{k}}$ are the subuniverses of an algebra \mathcal{K}_k on R_k called *Krasner algebra* (of the first kind [14]).

We choose the following intuitive basic operations: a binary \sqcap , four unary $\zeta, \tau, \Delta, \nabla$ and one nullary ω_k . Let $\rho \in R_{\mathbf{k}}^{(h)}$ and $\sigma \in R_k^{(l)}$.

Define $\rho \sqcap \sigma := \rho \cap \sigma$ when h = l and $\rho \sqcap \sigma := \emptyset \in R_k^{(1)}$ otherwise (formally we distinguish the empty relations of different arities; with ζ, τ, Δ and ∇ we can change their arities at will and hence it poses no real problem).

For k = 1 set $\alpha(\rho) = \rho$ for all $\alpha \in \{\zeta, \tau, \Delta\}$ while for $k \geq 2$

$$\zeta(\rho) := \{(a_2, \dots, a_h, a_1) : (a_1, \dots, a_h) \in \rho\},$$

$$\tau(\rho) := \{(a_2, a_1, a_3, \dots, a_h) : (a_1, \dots, a_h) \in \rho\},$$

$$\Delta(\rho) = \{(a_1, \dots, a_{h-1}) : (a_1, \dots, a_h) \in \rho \text{ for some } a_h\}.$$

Next

$$\nabla(\rho) := \{(a_1, \dots, a_{h+1}) : (a_1, \dots, a_h) \in \rho\}.$$

Finally, set $\omega_k = \{(x, x) : x \in \mathbf{k}\}$ (the identity or the least equivalence relation on \mathbf{k}). Now the *Krasner algebra* (of type 2,1,1,1,1,0) is

$$\mathcal{K}_k = \langle R_k; \sqcap, \zeta, \tau, \Delta, \nabla, \omega_k \rangle$$

For $f \in O_k$ set

$$Inv f = \{ \rho \in R_k : f \text{ preserves } \rho \}.$$

The Galois closed subsets of R_k are $\text{Inv}X = \bigcap_{f \in X} \text{Inv}f$ with $X \subseteq O_k$. The following basic results from [1,3] demonstrate the relevance of Krasner algebras in our study of the lattice \mathcal{L}_k .

Proposition 1

- (i) $\{\operatorname{Inv} X : X \subseteq O_k\} = \operatorname{Sub}(\mathcal{K}_k),$
- (ii) $C \mapsto \operatorname{Inv} C$ is a lattice antiisomorphism from \mathcal{L}_k onto $\operatorname{\mathbf{Sub}}(\mathcal{K}_k)$ and
- (iii) $\operatorname{Sub}(\mathcal{K}_k)$ is lattice isomorphic to \mathcal{L}_k^d .

In (ii) above it is assumed that the map is an onto bijection such that $C \subset D \iff \text{Inv}C \supset \text{Inv}D$ for all $C, D \in \mathcal{L}_{\mathbf{k}}$.

The subuniverses of \mathcal{K}_k are called the *coclones* on **k**.

Notations for the Clones in the Post Lattice

To present the results on generation of the clone lattice by means of its irreducible elements and to demonstrate some results obtained in the Boolean case we here present the notation of the Post Lattice of Boolean clones used in recent publications, particularly in [11].

First, to conform with the above mentioned publication we use the notation Ω for the set of all Boolean functions, i.e. $\Omega = O_2$. For any subset $A \subseteq \Omega$, A(n) denotes the set of all n-variable functions in A.

For $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ we write $\alpha \leq \beta$ if $a_i \leq b_i$ for all $i \in \{1, \ldots, n\}$, $\alpha \wedge \beta = (min(a_1, b_1), \ldots, min(a_n, b_n))$, $\alpha \vee \beta = (max(a_1, b_1), \ldots, max(a_n, b_n))$ and $\bar{\alpha} = (\bar{a}_1, \ldots, \bar{a}_n)$, where $\bar{a}_i = 1 - a_i$.

We say that for n > 1 and $1 \le i \le n$ a function $g \in \Omega(n)$ is obtained from a function $f \in \Omega(n-1)$ by the *introduction of the i-th dummy variable* if $g(x_1, \ldots, x_n) \approx f(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

Here and in the sequel \approx denotes an identity, i.e. the validity for all $a_1, \ldots, a_n \in E$.

Two functions f and g are called equivalent if both can be obtained from a common function by introduction of a finite number of dummy variables. We write $f \approx g$ for equivalent functions.

We use the following notations for elementary Boolean operations:

The four operations in $\Omega(1)$: $\mathbf{0}$ - constant 0; $\mathbf{1}$ - constant 1; x (or Id) - identity map and \bar{x} - negation. Some dyadic operations: xy or $x \wedge y$ - conjunction; $x \vee y$ - disjunction; $x \oplus y$ - mod2 addition; $x \to y$ - implication; $x \sim y$ - equivalence; x/y - Sheffer stroke; $x \downarrow y$ - Pierce arrow.

For a set of functions A we denote by [A] the closure of A, i.e. the least closed class (see above) containing A. Thus, $[\{x \oplus y, x \lor y, \mathbf{1}\}]$ denotes the closure of $\{x \oplus y, x \lor y, \mathbf{1}\}$, i.e. the closed class generated by operations $x \oplus y, x \lor y$ and $\mathbf{1}$.

Below we give the notation and description of each clone of Boolean function.

```
\Omega: All Boolean functions.
```

 T_0 : Functions which preserve the constant 0, thus $f \in T_0 \Leftrightarrow f(0,\ldots,0) = 0$.

$$T_1: T_1 = \{f \in \Omega | f(1,\ldots,1) = 1\}.$$

 T_c : Idempotent functions. $T_c = T_0 \cap T_1$.

M: Monotone (positive) functions.

$$f \in M \Leftrightarrow f(\alpha) \leq f(\beta)$$
 whenever $\alpha \leq \beta$.

$$M_0: M_0 = M \cap T_0.$$

$$M_1: M_1 = M \cap T_1.$$

$$M_c: M_c = M \cap T_c.$$

S: Selfdual functions.
$$S = \bigcup_{n=1}^{\infty} S(n); S(n) = \{ f \in \Omega(n) | f(\bar{x}_1, \dots, \bar{x}_n) \approx \bar{f}(x_1, \dots, x_n) \}.$$

$$S_c: S_c = S \cap T_c.$$

$$S_m: S_m = S \cap M.$$

$$\begin{array}{c} L: \ Linear \ {\rm functions.} \ L=\bigcup_{n=0}^{\infty}L(n); \ L(n)=\\ \{a_0\oplus a_1x_1\oplus\cdots\oplus a_nx_n|\ a_0,\ldots,a_n\in E\}. \end{array}$$

$$L_0: L_0 = L \cap T_0.$$

$$L_1: L_1 = L \cap T_1.$$

$$L_s$$
: $L_s = L \cap S$.

$$L_c$$
: $L_c = L \cap T_c$.

$$\Lambda_c$$
: Conjunctions. $\Lambda_c = \{x_1 \wedge \cdots \wedge x_n | n \geq 1\}.$

$$\Lambda_0: \Lambda_0 = \Lambda_c \cup C_0.$$

$$\Lambda_1 : \Lambda_1 = \Lambda_c \cup C_1.$$

$$\Lambda : \Lambda = \Lambda_c \cup C.$$

$$V_c$$
: Disjunctions. $V_c = \{x_1 \vee \cdots \vee x_n | n \geq 1\}.$

$$V_0: V_0 = V_c \cup C_0.$$

$$V_1 : V_1 = V_c \cup C_1.$$

$$V: V = V_c \cup C$$
.

$$U^i: U^i = \{f \in \Omega | \forall \alpha_1, \dots, \alpha_i \in E^n : f(\alpha_1) = \dots = f(\alpha_i) = 1 \Rightarrow \alpha_1 \wedge \dots \wedge \alpha_i \neq (0, \dots, 0)\}, i = 2, 3, \dots; n \text{ is defined by } f \in \Omega(n). U^2 \text{ is sometimes referred to as the class of clique functions, see e.g., [13].}$$

$$U^{\infty}: U^{\infty} = \bigcap_{i=2}^{\infty} U^{i}.$$

$$U_m^i : U_m^i = M \cap U^i, i = 2, 3, \dots$$

$$U_m^{\infty}$$
: $U_m^{\infty} = M \cap U^{\infty}$.

$$U_c^i : U_c^i = T_c \cap U^i, i = 2, 3, \dots$$

$$U_c^{\infty}: U_c^{\infty} = T_c \cap U^{\infty}.$$

$$U_{mc}^{i}: U_{mc}^{i} = M_{c} \cap U^{i}, i = 2, 3, \dots$$

$$U_{mc}^{\infty}: U_{mc}^{\infty} = M_c \cap U^{\infty}.$$

$$W^i: W^i = \{f \in \Omega | \forall \alpha_1, \dots, \alpha_i \in E^n: f(\alpha_1) = \dots = f(\alpha_i) = 0 \Rightarrow \alpha_1 \vee \dots \vee \alpha_i \neq (1, \dots, 1)\}, i = 2, 3, \dots; n \text{ is defined by } f \in \Omega(n). [W^2 - co-clique functions]$$

$$W^{\infty}: W^{\infty} = \bigcap_{i=2}^{\infty} W^{i}.$$

$$W_m^i: W_m^i = M \cap W^i, i = 2, 3, \dots$$

$$W_m^{\infty}: W_m^{\infty} = M \cap W^{\infty}.$$

```
\begin{split} W_c^i : \ W_c^i &= T_c \cap W^i, \ i = 2, 3, \dots \\ W_c^\infty : \ W_c^\infty &= T_c \cap W^\infty. \\ W_{mc}^i : \ W_{mc}^i &= M_c \cap W^i, \ i = 2, 3, \dots \\ W_{mc}^\infty : \ W_{mc}^\infty &= M_c \cap W^\infty. \\ I_c : \ I_c &= \{ f \in \Omega | f \equiv x \}. \\ I_0 : \ I_0 &= I_c \cup C_0. \\ I_1 : \ I_1 &= I_c \cup C_1. \\ I : \ I &= I_c \cup C. \\ I^* : \ I^* &= I_c \cup \{ f \in \Omega | f \equiv \bar{x} \}. \\ \Omega(1) : \ Monadic \ operations. \ \Omega(1) &= C \cup I^*. \end{split}
```

All clones (and closed classes) were completely described by E.Post in 1941 [15]. They form the lattice \mathcal{L}_2 presented in Figure 1, which is often referred to as the Post Lattice.

Results on Irreducible Clones

In this section we present the major results obtained so far in the study of irreducible clones. First we affirm the generating power of such clones and show the optimal generating expressions for the Post Lattice. Next, for the general case of finite valued logic algebra, we provide criteria, examples and enumerating results for both join- and meet-irreducible clones. Finally, we focus our attention on the monoidal part of the lattice and show some complete results for the join-irreducible monoids.

Irreducible Elements as Lattice Generators. Post Lattice

Using Post's description [15] of \mathcal{L}_2 the paper [10] completely determined the sets \mathbf{J}_2 and \mathbf{M}_2 both consisting of two infinite (correspondingly descending and ascending) chains in \mathcal{L}_2 together with some additional clones.

Without much effort it was shown in [11] that the lattice \mathcal{L}_2 is generated by means of meet and join operations over either kind of irreducible clones. Later it was confirmed that the generating ability of such clones also holds for any \mathcal{L}_k , which due to the lattice's algebraic and dually algebraic character. Note that this statement can not be generalized to every lattice \mathcal{L}_A , and there are known examples of lattices with an element which cannot be expressed as a join of join-irreducibles nor as a meet of meet-irreducible elements.

Before proceeding to generating expressions let us observe a structural property of irreducible elements in \mathcal{L}_2 .

Observation 1 Both subgraphs of \mathcal{L}_2 – one induced by all join-irreducible clones and the other one by all meet-irreducible clones are trees, i.e., connected acyclic graphs.

It was a simple yet appealing question whether the observation above holds for any \mathcal{L}_k . The proposition below ([10]) negatively answers this question.

Proposition 2 The subgraphs of \mathcal{L}_k induced by all join-irreducible clones is not a connected graph when $k \geq 5$.

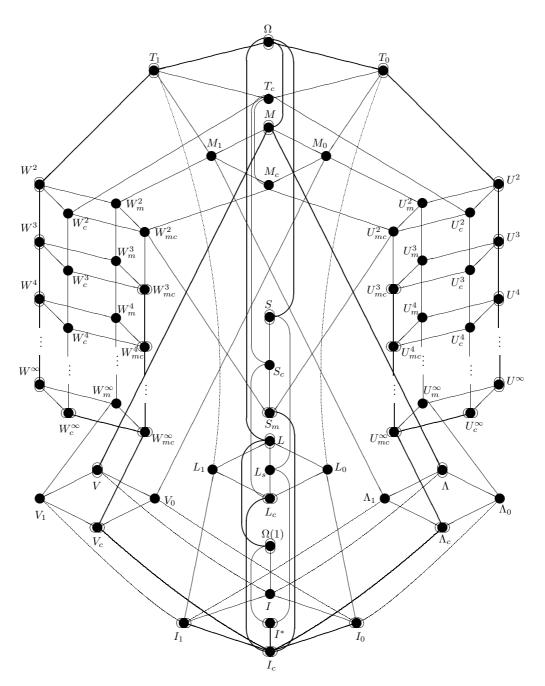


Рис. 1: \mathcal{L}_2 – the Post Lattice.

Clone	V-term	∧-term	min-length
Ω	$I^* \vee \Lambda_c$	Ω	1
T_0	$I_0 \lor U_c^\infty$	T_0	1
T_1	$I_1 \vee W_c^{\infty}$	T_1	1
T_c	$U_c^{\infty} \vee W_c^{\infty}$	$T_0 \wedge T_1$	2
$\stackrel{\circ}{M}$	$I_0 \lor I_1 \lor U_{mc}^{\infty}$	M	1
M_0	$I_0 \lor V_c \lor U_{mc}^{\infty}$	$T_0 \wedge M$	$\stackrel{ ext{-}}{2}$
M_1	$I_1 \vee \Lambda_c \vee W_{mc}^{\infty}$	$T_1 \wedge M$	$\frac{1}{2}$
M_c	$V_c ee \Lambda_c$	$T_0 \wedge T_1 \wedge M$	2
S	$I^* \lor S_m$	S	1
$\overset{\sim}{S_c}$	$L_c ee S_m$	$T_0 \wedge S$	2
S_m	S_m	$S \wedge M$	$\frac{1}{2}$
L	$I_0 \vee I_1 \vee L_c$	L	1
L_0	$I_0 \lor L_c$	$T_0 \wedge L$	$\frac{1}{2}$
L_1	$I_1 \lor L_c$	$T_1 \wedge L$	$\frac{1}{2}$
L_c	L_c	$T_0 \wedge T_1 \wedge L$	1
L_s	$I^* ee L_c$	$L \wedge S$	$\frac{1}{2}$
Λ	$I_0 ee I_1 ee \Lambda_c$	Λ	1
Λ_0	$I_0 \vee I_1 \vee I_C$ $I_0 \vee \Lambda_c$	$T_0 \wedge \Lambda$	$\frac{1}{2}$
Λ_1	$I_1 \lor \Lambda_c$	$T_1 \wedge \Lambda$	$\frac{2}{2}$
Λ_c	Λ_c	$T_0 \wedge T_1 \wedge \Lambda$	1
V	$I_0 \lor I_1 \lor V_c$	V	1
V_0	$I_0 \lor I_1 \lor V_c$	$T_0 \wedge V$	$\overset{1}{2}$
V_1	$I_1 \lor V_c$	$T_1 \wedge V$	$\frac{2}{2}$
V_c	V_c	$T_0 \wedge T_1 \wedge V$	1
U^2	$I_0 \vee U_c^{\infty} \vee U_{mc}^{3} \vee S_m$	U^2	1
$U^i, i \ge 3$	$I_0 \lor U_c^c \lor U_{mc}^i \lor U_m^i$	U^i	1
U^{∞}	$I_0 \lor U_c \lor U_{mc}$	U^{∞}	1
W^2	$I_0 \vee U_c^{\infty}$ $I_1 \vee W_c^{\infty} \vee W_{mc}^3 \vee S_m$	W^2	1
$W^i, i \geq 3$	$I_1 \vee W_c^{c} \vee W_{mc}^{i} \vee S_m^{i}$	W^i	1
W^{∞}	$I_1 \vee W_c \vee W_{mc}$ $I_1 \vee W_c^{\infty}$	W^{∞}	1
U_c^2	$U_c^{\infty} \vee U_{mc}^{3} \vee S_m$	$U^2 \wedge T_1$	$\overset{1}{2}$
$U_c^i, i \ge 3$	$U_c^{\infty} \vee U_{mc}^i \vee U_m^i$	$U^i \wedge T_1$	2
U_c^{∞}	U_c^{∞}	$U^{\infty} \wedge T_1$	1
W_c^2	$W_c^{\infty} \vee W_{mc}^3 \vee S_m$	$W^2 \wedge T_0$	$\frac{1}{2}$
$W_c^i, i \geq 3$	$W_c^{\infty} \vee W_{mc}^{\infty} \vee U_{mc}^{m}$	$W^i \wedge T_0$	2
$W_c, i \ge 3$ W_c^{∞}	$W_c \forall W_{mc} \\ W_c^{\infty}$	$W^{\infty} \wedge T_0$	1
$U_m^{v_c}$	$I_0 \vee \Lambda_c \vee S_m$	$U^2 \wedge M$	$\frac{1}{2}$
$U_m^i, i \ge 3$	$I_0 \vee \Lambda_c \vee S_m$ $I_0 \vee \Lambda_c \vee U_{mc}^i$	$U^i \wedge M$	$\frac{2}{2}$
U_m^{∞}	$I_0 \lor I_c \lor U_{mc}$ $I_0 \lor U_{mc}^{\infty}$	$U^{\infty} \wedge M$	$\frac{2}{2}$
W_m^2	$I_1 \lor V_c \lor S_m$	$W^2 \wedge M$	$\frac{2}{2}$
$W_m^i, i \geq 3$	$I_1 \vee V_c \vee S_m$ $I_1 \vee V_c \vee W_{mc}^i$	$W^i \wedge M$	$\frac{2}{2}$
$W_m, t \ge 3$ W_m^{∞}	$I_1 \lor V_c \lor W_{mc}$ $I_1 \lor W_{mc}^{\infty}$	$W^{\infty} \wedge M$	$\frac{2}{2}$
$U_{mc}^{N_{m}}$	$U_{mc}^{\infty} \vee S_m$	$U^2 \wedge M \wedge T_1$	$\frac{2}{2}$
$U_{mc}^{i}, i \ge 3$	U_{mc}^{i}	$U^i \wedge M \wedge T_1$	1
II^{∞}	U_{mc}^{∞}	$U^{\infty} \wedge M \wedge T_1$	1
U_{mc}^{∞} W_{mc}^{2}	$W_{mc}^{\infty} \lor S_m$	$W^2 \wedge M \wedge T_0$	2
$W_{mc}^{i}, i \ge 3$	$W_{mc} \lor S_m \ W_{mc}^i$	$W \wedge M \wedge T_0$ $W^i \wedge M \wedge T_0$	1
$W_{mc}, t \ge 3$ W_{mc}^{∞}	W_{mc}^{∞}	$W \wedge M \wedge T_0$ $W^{\infty} \wedge M \wedge T_0$	1
$\Omega(1)$	$I^* \vee I_0$	$\Omega(1)$	1
I^*	$I^* \stackrel{I imes I_0}{=} I^*$	$\Omega(1) \wedge S$	1
			ļ.
I_0	I_0	$\Omega(1) \wedge T_0$	1 1
I_1	I_1	$\Omega(1) \wedge T_1$	1
I_c	I_c	$\Omega(1) \wedge T_0 \wedge T_1$	1

 Table 1. Representation of Clones

Now, let us consider clones that belong to $\mathbf{J}_k \cap \mathbf{M}_k$, i.e totally irreducible by both join and meet operation. We call such elements of \mathcal{L}_k transit clones.

Another simple observation states:

Observation 2 There are no transit clones in \mathcal{L}_2

The problem of transit clones still remains open for the general case of \mathcal{L}_k for $k \geq 3$, and it is not known whether such clones exist. However, we have recently shown that there are no transit clones among monoids in \mathcal{L}_k (we will mention about such problems again in our concluding remarks).

Finally, we present a complete set of optimal expressions that demonstrate the generation of \mathcal{L}_2 . An expression which is the join (meet) of some join-(meet-)irreducible clones is referred to as a \vee -term (\wedge -term).

Table 1, presents every clone of \mathcal{L}_2 in both a \vee -term and a \wedge -term. All expressions can be verified by tracing the relations on the Figure 1.

Theorem 1 The following hold for \mathcal{L}_2

- 1. Every clone except U^2 and W^2 is the join of at most three join-irreducible clones. Both U^2 and W^2 are the join of four join-irreducible clones;
 - **2.** Every clone is the meet of at most three meet-irreducible clones;
- **3.** Every clone is either the join of at most two join-irreducible clones, or a meet of at most two meet-irreducible clones.

Join-Irreducible Clones

The following very simple and probably known proposition characterizes the totally join-irreducible elements of the lattice $\mathbf{Sub}(\mathcal{A})$.

Proposition 3 Let A be an algebra and let $X \in Sub(A)$ be not the least element of Sub(A). Then X is a totally join-irreducible element of Sub(A) if and only if

- (i) X is single generated and
- (ii) the set Z of all those elements of X that are not single generators of X is a subuniverse of A.

Based on proposition above we characterize J_A for any set A. For clone C denote by S_C the set of of its single generators (i.e., $f \in C \iff [f] = C$) and set $Z_C = C \setminus S_C$.

Proposition 4 The following are equivalent for a clone C on A distinct from the least clone Q_A :

- (1) C is totally join-irreducible;
- (2) S_C is nonvoid and Z_C is a clone;
- (3) S_C is nonvoid and Z_C is closed under *.

Define a binary relation \Diamond on O_A by setting $f \Diamond g$ if [f] = [g]. It is immediate that \Diamond is an equivalence relation on O_A . Denote by ε_A the cardinality of the set of the blocks of \Diamond on A.

Corollary 1 (i) $|\mathbf{J}_A| \leq \varepsilon_A$,

- (ii) $|\mathbf{J}_A| \leq |A|^{|A|}$ for A infinite, and
- (iii) $|\mathbf{J_k}| \leq \aleph_0$ for every integer $k \geq 2$ ([10]).

Next, we demonstrate embedding of certain totally join-irreducible clones.

Let $\emptyset \neq A \subset B$. In this section we produce a lattice embedding of \mathcal{L}_A into \mathcal{L}_B mapping certain join-irreducible clones of \mathcal{L}_A onto certain join-irreducible clones of \mathcal{L}_B . From this we obtain that for every $k \geq 2$ the set $J_{\mathbf{k}}$ of join-irreducible clones is countably infinite.

We use the following well-known map (see e.g. [5]). Let φ be a fixed retraction from B onto A, i.e., $\mathbf{im}\varphi=A$ and $\varphi(\varphi(b))=\varphi(b)$ for all $b\in B$. To $f\in O_A^{(n)}$ assign $f^+\in O_B^{(n)}$ defined by setting for all $b_1,\ldots,b_n\in B$

$$f^{+}(b_1,\ldots,b_n) = f(\varphi(b_1),\ldots,\varphi(b_n)). \tag{1}$$

In view of $\mathbf{im}\varphi = A$ clearly $\psi : f \mapsto f^+$ is well defined. Extend the map ψ to the subset of O_A in the usual way: $X^+ = \{f^+ : f \in X\}$ for all $X \subseteq O_A$. Recall from subsection 1.2 that the iterative algebra on a set C is $\mathcal{M}'_C = \langle O_C; *, \zeta, \tau, \Delta, \nabla \rangle$.

Lemma 1 Let A, B, φ and ψ be as above. Then:

- (i) ψ is an embedding of \mathcal{M}'_A into \mathcal{M}'_B , and
- (ii) the extended map ψ is a lattice isomorphism of \mathcal{L}_A onto the interval $[Q_A^+, O_A^+]$ of $\mathbf{Sub}(\mathcal{M}_B')$.

The map ψ maps only certain join-irreducible clones on A onto join-irreducible clones on B. Call a clone C on A tout if C is the closed class on A generated by the set $C \setminus Q_A$; in other words if $e_{A,1}^1 (= \mathbf{id}_A)$ can be obtained as a composition of operations from C other than projections.

Denote by T_A the set of all tout totally join-irreducible clones on A. Finally, for a clone C on A set $C^0 = C^+ \cup Q_B$ and notice that C^0 is a clone on B since C^+ is a closed class on B by Lemma 1.

Proposition 5 Let A, B, φ and ψ be as above. Then ψ embeds the poset (T_A, \subseteq) into the poset (J_B, \subseteq) of totally join-irreducible clones on B.

Remark 1. In Proposition 5 the assumption of toutness is essential. Indeed, if C is a join-irreducible clone on A that is not tout and D is the clone covered by C then C^0 is the join of D^0 and $(C \setminus Q_A)^0$.

Corollary 2 (i) In $\mathcal{L}_{\mathbf{k}}$ there exist 2^{k-1} chains of totally join-irreducible clones such that: a) each chain is of type $\omega^* + 2$ and is non-refinable, b) clones in different chains are incomparable and c) the least clone in each chain is a minimal clone (i.e., an atom of $\mathcal{L}_{\mathbf{k}}$);

(ii) The set J_k of totally join-irreducible clones of \mathcal{L}_k is countably infinite.

According to [10] there are two such chains in \mathcal{L}_2 . One is

$$MTW_3 \supset \cdots \supset MTW_\infty \supset \Lambda \supset Q_2$$
 (2)

and the other one is dual to it (see subsection 1.4 or [10] for the notations). The clones are tout since $e_{2,1}^1(x) = x \wedge x = x$ for all $x \in \mathbf{2}$.

Complete proofs of most of the statements in this subsection can be found in [12].

Meet-Irreducible Clones

We characterize the totally meet-irreducible clones on \mathbf{k} . For a clone C on \mathbf{k} set

$$R_C = \{ \rho \in R_{\mathbf{k}} : \text{Pol}\rho = C \}; \ U_C = \text{Inv}C \setminus R_C.$$

Recall that C is rational if $C = \text{Pol}\rho$ for some $\rho \in R_k$.

Proposition 6 The following are equivalent for a clone C on k distinct from Q_k :

- 1) C is a totally meet-irreducible clone,
- 2) C is rational and U_C is a subuniverse of $\mathcal{K}_{\mathbf{k}}$,
- 3) C is rational and U_C is closed under Δ and \Box .

Next we construct chains of totally meet-irreducible clones on ${\bf k}$.

Let A be a proper subset of **k**. For $h \ge 1$ set

$$\rho_h^A = k^h \setminus A^h;$$

in other words ρ_h^A is the set of h-tuples over **k** with at least one coordinate outside A. It will be convenient to set $\rho_0^A = \emptyset$.

Example 2.

1) Let k=2. Then for $h\geq 1$

$$\rho_h^{\{0\}} = \mathbf{2}^h \setminus \{(0, \dots, 0)\},$$

$$\rho_h^{\{1\}} = \mathbf{2}^h \setminus \{(1, \dots, 1)\}.$$

The corresponding clones $\operatorname{Pol}_h^{\{0\}}$ and $\operatorname{Pol}_h^{\{1\}}$ are the clones U_h and W_h of the Post Lattice [15].

2) Let k = 3. Consider

$$\rho_h^{\{0\}} = \mathbf{3}^h \setminus \{(0, \dots, 0)\},$$
$$\rho_h^{\mathbf{2}} = \mathbf{3}^h \setminus \mathbf{2}^h.$$

Note that $\rho_1^A = \mathbf{k} \setminus A$ and therefore $\operatorname{Pol}\rho_1^A$ is a maximal clone (a coatom of \mathcal{L}_3 [19]) and thus is totally meet-irreducible. Notice that $O_k = \operatorname{Pol}\emptyset$ is the unique cover of $\operatorname{Pol}\rho_1^A$.

Theorem 2 If A is a proper subset of **k** then for every $h \ge 1$ the clone Pol_h^A is totally meet-irreducible and $\operatorname{Pol}_{h-1}^A$ is its unique cover in $\mathcal{L}_{\mathbf{k}}$.

The following are immediate corollaries of the theorem.

Corollary 3 (i) For every proper subset A of k

$$\operatorname{Pol}\rho_1^A \supset \operatorname{Pol}\rho_2^A \supset \cdots$$

is an unrefinable chain C_A of totally meet-irreducible clones in $\mathcal{L}_{\mathbf{k}}$,

- (ii) there are $2^k 2$ chains C_A , and
- (iii) $\operatorname{Pol}\rho_h^A$ and $\operatorname{Pol}\rho_{h'}^{A'}$ are incomparable whenever $\emptyset \neq A, A' \subset \mathbf{k}, A \neq A'$ and $h', h \geq 1$.

Corollary 4 The set \mathbf{M}_k of totally meet-irreducible clones on \mathbf{k} is countably infinite.

Details of the construction above and the proofs of major statements in this part can be found in [12].

Irreducible Monoids

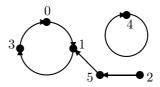
Here we focus on the monoidal part of the lattice $\mathcal{L}_{\mathbf{k}}$, i.e., clones which contain only one variable functions. In our early paper [10] we characterized all join-irreducible monoids in $\mathcal{L}_{\mathbf{k}}$.

Let $f \in O_k^{(1)}$, i.e. f is a one variable function over \mathbf{k} . For f we denote $\mathbf{k}_0 := \mathbf{k}$, and recursively $\mathbf{k}_{n+1} := f(\mathbf{k}_n)$ for $n \ge 0$.

We proceed with some facts and further notations:

- There is and integer $t, t \ge 0$, such that $\mathbf{k}_{t+1} = \mathbf{k}_t$. We denote the smallest such \mathbf{k}_t by \mathbf{k}_f ;
- $f|\mathbf{k}_f$ is a permutation on \mathbf{k}_f , and therefore can be presented as a product of cyclic permutations, called the *cycles* of f;
- An element of a of $\mathbf{k} \setminus \mathbf{k}_f$ is called a *tail* of f. The smallest integer s such that $f^s(a) \in \mathbf{k}_f$ is called the *length* of the tail a;
- The order of f, i.e. the smallest positive integer s such that $f^s|\mathbf{k}_f = Id|\mathbf{k}_f$, is equal to the least common multiplier of the orders of cycles of f.

Below is an example of graphical interpretation of a function:



Vertices (5) and (2) correspond to tails of length 1 and 2, respectively.

Using the above terminology we can characterize the monoidal atoms of the lattice which are generated by a single one variable function:

Lemma 2 A clone $C = [\{f\}]$ is an atom if and only if it satisfies one of the following conditions:

- (1) The function f has some tails of length 1, but has no tails of length greater than 1, and $f|\mathbf{k}_f = Id|\mathbf{k}_f$;
- (2) the function f has no tails and its order is a prime number.

The following two lemmas lead to the main theorem below which fully characterizes the join-irreducible monoids.

Lemma 3 If f has a tail of length greater than one then the clone $C = [\{f\}]$ is totally join-irreducible. generated by a single one variable function:

Lemma 4 If f has no tails of length greater than one, then $C = [\{f\}]$ is totally join-irreducible if and only if the order of f is a power of a prime.

Theorem 3 Consider a monoidal clone $C = [\{f\}]$ generated by one variable function f. C is totally join-irreducible if and only if its generator f satisfies one the following conditions:

- (1) f has a tail of length greater than one;
- (2) f has no tails of length greater than one, and its order is a power of a prime.

Corollary 5 All singleton generated monoids in k-valued logic are totally join-irreducible when k < 5.

As seen above there are many join-irreducible clones in the monoidal part of the lattice and they can be well classified by their single generators. The questions regarding the meet-irreducible monoids, including the mere existence of such remains open.

Concluding Remarks

To the best knowledge of the author of this report the study of irreducible clones began in mid 90's of the past century. Since then, as shown in this presentation, a considerable progress has been made in understanding, enumeration and characterization of such clones. Meanwhile, in spite of a certain 'compactness' of the set of such clones (their number is countable in a continuum element lattice) and known criteria and examples of chains, their full description remains to be an extremely difficult task.

Below we pose some potentially solvable questions which remain open to this day:

- 1. The question of the existence of transit clones was raised since our first publications and talks [10,9]. Such clone is both single generated and rational by Propositions 3 and 6. Recently the author proved (to be published soon) that there are no transit clones among the monoids, but the question remains open for the lattice \mathcal{L}_k when $k \geq 3$.
- 2. The two ω^* -chains of totally join-irreducible clones on **2** consist of rational clones. Indeed, for $h \geq 3$ the clone $MTV_h = \text{Pol}\rho_h$, where

$$(a_1, \dots, a_{h+3}) \in \rho_h \iff (a_1, \dots, a_h) \neq (0, \dots, 0),$$

$$a_{h+1} = 0, (a_{h+2}, a_{h+3}) \neq (1, 0).$$

A natural question is whether every such clone on \mathbf{k} is also rational.

Let k > 2 and let φ be a retraction from \mathbf{k} onto $\mathbf{2}$. Set $D = Q_{\mathbf{k}} \cup \{f \in O_{\mathbf{k}} : \mathbf{im} f \subseteq \mathbf{2}\}$. It is easy to see that D is a clone on \mathbf{k} . Further set $\sigma = \{(x, \varphi(x)) : x \in \mathbf{k}\}$. We show

$$D \cap \operatorname{Pol}\sigma = Q_{\mathbf{k}} \cup O_{\mathbf{2}}^+,$$

where f^+ was defined in Subsection 2.2. To show the \subseteq inclusion let $f \in (D \cap \text{Pol}\sigma) \setminus Q_k$ be n-ary. Then $\text{im } f \subseteq \mathbf{2}$.

Denote by g the restriction of f to $\mathbf{2}$. Clearly $g \in O_{\mathbf{2}}^{(n)}$. We show that $g^+ = f$. Indeed, let $(a_1, \ldots, a_n) \in \mathbf{k}^n$ be arbitrary. As $f \in \text{Pol}\sigma$ and $\varphi(x) = x$ on $\mathbf{2}$ we get

$$f(a_1, \dots, a_n) = \varphi(f(a_1, \dots, a_n)) =$$

$$= f(\varphi(a_1), \dots, \varphi(a_n)) = g(\varphi(a_1), \dots, \varphi(a_n)),$$

proving $f = g^+ \in O_2^+$.

The proof of the \supseteq part is similar.

It can be shown that the *n*-ary graphics of the clone $D \cap \text{Pol}\sigma$ satisfy $\Gamma_1 \supset \Gamma_1 \supset \cdots$ and thus $D \cap \text{Pol}\sigma = \bigcap_{i>1} \Gamma_i$ is not rational.

- 3. Among the join-irreducible clones the atoms of $\mathcal{L}_{\mathbf{k}}$, called minimal clones, are of interest. They are completely known for k=2 [15] and for k=3 [2], while for k=4 only the case of minimal clones generated by semiprojections is not yet settled [14,18].
- 4. Among the totally meet-irreducible clones also are the essentially minimal clones [6,7] which cover exactly one essentially unary clone distinct from $Q_{\mathbf{k}}$.
- 5. Among the totally meet-irreudcible clones on **k** the submaximal clones are of interest. They are clones covered by exactly one maximal clone and are explicitly known for k = 2 [15] and k = 3 [5].
- 6. We pose a question: Let C be a totally meet-irreducible clone on \mathbf{k} . When is U_C not single generated, i.e., the unique cover of C not rational?

Acknowledgement. The author expresses gratitude to his collaborators, particularly Professors Akihiro Nozaki and Ivo G. Rosenberg for their interest and valuable contributions regarding the irreducible clones.

The author dedicates this report to Prof. Valery B. Kudryavcev extending best wishes on the occasion of his 70-th birthday.

The Bibliography

- 1. V. B. Bodnarchuk, L. A. Kalužnin, V. N. Kotov, B. A. Romov, "Galois theory for Post algebras," *Kibernetika (Kiev)*, Part I: #3, pp. 1-10, 1969, Part II: #5, pp. 1-9, 1969
 - 2. B. Čsakany, "All minimal clones on the three-element set," Acta Cybernetica #6, pp. 227-238, 1983
 - 3. D. Geiger, "Closed Systems of Functions and Predicates," Pacific J. Math. #27, pp. 95-100, 1968
- 4. A. V. Kuznetsov. "Lattices with Closure and Criteria for Functional Completeness" [Russian], $Uspekhi\ Mat.\ Nauk\ \#16/2(98),\ pp.\ 201-202,\ 1961$
- 5. D. Lau, "Submaximale Klassen von P_3 ," Elektronische Informationsverarbeitung und Kybernetik, 21(1/2), pp. 9-22, 1985
- 6. H. Machida, I.G. Rosenberg, "Classifying Essentially Minimal Clones," *Proceedings of the 14th Intern. Symp. on Multiple-Valued Logic*, IEEE Press, pp. 4-7, 1984
- 7. H. Machida, I.G. Rosenberg, "Essentially Minimal Groupoids," *Algebras and Orders*, Kluwer Acad., NATO ASI Series, C, vol. 389, pp. 287-316, 1991
- 8. A. I. Mal'tsev, "Iterative Post Algebras and Varieties" [Russian], Algebra i Logika, #6, pp. 5-24, 1966
- 9. G. Pogosyan, "On Irreducible Clones," Winter-1999 Meeting of Canadian Mathematical Society. Presentation, Dec.1999
- 10. G. Pogosyan, A. Nozaki "Join-Irreducible Clones of Multiple-Valued Logic Algebra," *Proceedings of the 25th Intern. Symp. on Multiple-Valued Logic*, IEEE Press, pp. 194-199, 1995
- 11. G. Pogosyan, I.G. Rosenberg "Generation of the Post Lattice by Irreducible Clones," *Proceedings of the 33rd Intern. Symp. on Multiple-Valued Logic*, IEEE Press, pp. 309-314, 2003
- 12. G. Pogosyan, I.G. Rosenberg "Algebraic Properties of Totally Irreducible Elements of Clone Lattices," *Proceedings of the 34th Intern. Symp. on Multiple-Valued Logic*, IEEE Press, pp. 109-114, 2004
- 13. G. Pogosyan, M. Miyakawa, A. Nozaki, I.G. Rosenberg, "On the Number of Clique Boolean Functions," *IEICE Transactions*, Vol. E80-A, No.8, pp. 1502-1507, 1997
- 14. R. Pöschel, L.A. Kalužnin, "Funktionen- und Relationenalgebren," WEB Deutscher Verlag der Wissenschaften, Berlin, 1979
- 15. E. L. Post, "Two-Valued Iterative Systems of Mathematical Logic," *Annals of Math. Studies*, No.5, 1941, Princeton U. Press
- 16. I. G. Rosenberg, "Über die funktionale Vollständigkeit in dem mehrwertigen Logiken," Rozpravy Československe Akademie Věd. Ser. Math. Nat. Sci. , No.80, pp. 3-93, 1970, MR 45 #1732
- 17. I. G. Rosenberg, "On Galois Connection between Algebras and Relations and its Applications," Contributions of General Algebra, pp. 273-289, 1978
 - 18. A. Szendrei, "Clones in Universal Algebra," Les Presses de l'Univ. de Montréal, 1986

On determinant expansion and minor rank function of matrices over arbitrary Boolean algebra

Vladislav Poplavski

assistant professor, Faculty of Mechanics and Mathematics, Saratov State University, Astrakhanskaya 83, 410012 Saratov, Russia E-mail: PoplavskiVB@mail.ru

It is well known that for oriented semipermanents $\overset{+}{\nabla} A$, $\overset{-}{\nabla} A$ (component of bideterminant) and the permanent $PerA = \stackrel{+}{\nabla} A \cup \stackrel{-}{\nabla} A$ the Laplace's formulas, giving rows and columns expansion of any square matrix A with the elements over arbitrary commutative semiring, hold [1]-[3]. The Laplace's formulas for the right $RDetA = \stackrel{+}{\nabla} A \setminus \stackrel{-}{\nabla} A = \stackrel{+}{\nabla} A \cap (\stackrel{-}{\nabla} A)'$ and the left $LDetA = \stackrel{-}{\nabla} A \setminus \stackrel{+}{\nabla} A = \stackrel{-}{\nabla} A \setminus \stackrel{+}{\nabla} A = \stackrel{-}{\nabla} A \setminus \stackrel{-}{\nabla} A = \stackrel{-}{\nabla} A \cap \stackrel{-}{\nabla} A$ $\stackrel{-}{\nabla} A \cap (\stackrel{+}{\nabla} A)'$ oriented determinants of the square $n \times n-$ matrices A $(n \ge 2)$ over arbitrary Boolean algebra $\langle B, \cup, \cap, ', 0, I \rangle$, determined as corresponding Boolean differences of oriented semipermanents determined as $\overset{\pm}{\nabla} A = \bigcup_{(\alpha_1, \dots, \alpha_n) \in \overset{\pm}{P}} (a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \dots \cap a_n^{\alpha_n})$ (where a_j^i - the elements of matrix A, both

even and odd n-permutations of upper indices are defined as $\stackrel{+}{P}$, $\stackrel{-}{P}$ correspondingly) do not hold in a common case. The Laplace's formulas do not hold in a common case for the determinants determined as $DetA = RDetA \cup LDetA$. Attempts to define the conditions when expansions of such determinants are possible were undertaken earlier. The authors [4] give sufficient conditions of row expansion of the determinant. Nevertheless, the formulas of the expansion of determinants over the elements of rows or columns of the matrix with the elements from arbitrary Boolean algebra hold for a large class of matrices. The matrices which are further referred to as matrices without interiority are relative to such matrices.

Any Boolean matrix can be presented as $A = [(PerA)' \cap A] \cup [\triangle A \cap A] \cup [RDetA \cap A] \cup [LDetA \cap A]$. $\operatorname{Here}\triangle A = \stackrel{+}{\nabla} A \cap \stackrel{-}{\nabla} A$. Introducing the following designations $\check{A} = (\operatorname{Per} A)' \cap A$, $\hat{A} = \triangle A \cap A$, $\stackrel{+}{A} = A \cap A$ $RDetA \cap A$, $A = LDetA \cap A$ we write down the following linear combination of matrices with Boolean coefficients $A = \check{A} \cup \hat{A} \cup \overset{-}{A} \cup \overset{-}{A}$.

Definition 1. Call matrix \check{A} an exterior part and \hat{A} an interior part of matrix A. Call matrix $\overset{+}{A}\cup\overset{-}{A}$ a nondegenerate (determinate) part, consisting of positive $\overset{+}{A}$ and negative $\overset{-}{A}$ (oriented) parts correspondingly. Call matrix exterior, interior, positive, negative if it coincides with its corresponding exterior, interior, positive, negative part correspondingly.

The main properties of matrices \check{A} , \hat{A} , $\overset{+}{A}$, $\overset{-}{A}$ and conditions of unique presentation of an arbitrary square Boolean matrix as $A = \check{A} \cup \hat{A} \cup \overset{+}{A} \cup \overset{-}{A}$ can be found in [5]. Using the introduced here terminology in the case of a trivial Boolean algebra $B_2 = \{0, I\}$ any nonzero

Boolean $n \times n$ —matrix can be of any four types: exterior (defined by the condition $(\stackrel{+}{\nabla} A, \stackrel{-}{\nabla} A) = (0,0)$ or PerA = 0), interior (defined by the condition $(\stackrel{+}{\nabla} A, \stackrel{-}{\nabla} A) = (I, I)$ or $PerA = \triangle A = I$), positive (defined by the condition $(\stackrel{+}{\nabla} A, \stackrel{-}{\nabla} A) = (I,0)$ or PerA = RDetA = I)), it can be negative (defined by the condition $(\stackrel{+}{\nabla} A, \stackrel{-}{\nabla} A) = (0, I)$ or PerA = LDetA = I). **Definition 2.** The symbol $\partial_k^i A$ designates the $(n-1) \times (n-1)$ -matrix obtained from the $n \times n$ -

matrix A by removing an i-row and a k-column.

Theorem 1. Let matrix A over an arbitrary Boolean algebra is a matrix with zero interior part then formulas of an expansion of determinants over any i-row of this matrix hold:

$$RDetA = \bigcup_{k=1}^{n} (a_k^i \cap R_k^i Det(\partial_k^i A)),$$

where $R_k^i = R$ if i + k is even and $R_k^i = L$ if i + k is odd;

$$LDetA = \bigcup_{k=1}^{n} (a_k^i \cap L_k^i Det(\partial_k^i A)),$$

where $L_k^i = L$ if i + k is even and $L_k^i = R$ if i + k is odd.

The corollary is as follows

$$Det A = \bigcup_{k=1}^{n} (a_k^i \cap Det(\partial_k^i A)).$$

The similar statement is true for rows.

However, it does not follow from the condition of decomposability of determinants that matrices possess zero interiority. For example, matrix

$$J = \begin{pmatrix} I & \dots & I \\ \vdots & & \vdots \\ I & \dots & I \end{pmatrix}$$

is interior but the formulas of expansion given in Theorem 1 are true for it. Among 3×3 matrices there is the only one interior matrix over $B_2 = \{0, I\}$ for which formulas of expansion over any row or column are true and it is defined as the matrix J. It follows from the following theorem giving the problem a combinatorial character.

Definition 3. Call sequence $(a_1^{\alpha_1}, a_2^{\alpha_2}, \dots, a_n^{\alpha_n})$ an odd (even) diagonal of matrix A over $B_2 = \{0, I\}$ if $a_1^{\alpha_1} \cap a_2^{\alpha_2} \cap \dots \cap a_n^{\alpha_n} = I$ and n-permutation of upper indices $(\alpha_1, \alpha_2, \dots, \alpha_n)$ are odd (even correspondingly).

Theorem 2. For any interior matrix with elements from Boolean algebra $B_2 = \{0, I\}$ formulas of expansion over a row (or column) are true if and only if at least two diagonals, one odd and one even, pass or no diagonals at all pass through every nonzero element of this row. Besides there is at least one nonzero element of this row (or column) and there are at least two (odd and even) diagonals which pass through this nonzero element.

Example. Expansions over any row or column hold for an interior matrix

$$A = \begin{pmatrix} I & I & 0 & 0 \\ I & I & 0 & 0 \\ 0 & 0 & I & I \\ 0 & 0 & I & I \end{pmatrix}$$

as through its every nonzero element there pass at least one odd and one even diagonal.

Definition 4. Call k-minor of Boolean $m \times n$ -matrix $(1 \leq k \leq min(m,n))$ the determinant $Det M_{j_1,j_2,...,j_k}^{i_1,i_2,...,i_k}$ of the block $M_{j_1,j_2,...,j_k}^{i_1,i_2,...,i_k}$ constructed of entries of some k columns with numbers $j_1, j_2, ..., j_k$ and k rows with numbers $i_1, i_2, ..., i_k$ of this matrix.

Definition 5. A minor rank of a nonzero $m \times n$ -matrix A is called a natural number rankA = k $(1 \le k \le min(m, n))$ satisfying two conditions:

- 1) matrix has at least one square rank block $M_{k \times k}$ whose determinant is different from zero;
- 2) if matrix A has $(k+1) \times (k+1)$ -blocks, then their determinants are zero.

A rank of a zero matrix is considered to be zero.

As it has been mentioned above formulas of expansion of Boolean determinants over rows (columns) do not hold in common case, however, this does not interfere with fulfilling the following proposition.

Proposition. If a minor rank of matrix is equal to k then determinants of all its $(k+1) \times (k+1) - blocks$ and blocks of a higher dimension, if they exist, are zero.

Thus, the minor rank of matrix is the largest order of the square block whose determinant is different from zero.

The rows (columns) constructing a rank block $M_{k \times k}$ are called rank rows (rank columns). The following theorem is analogous to the base minor theorem for numerical matrices.

Theorem 3. For an arbitrary j-column $(row)(A)_j$ of the matrix A there are two linear combinations of rank columns $(rows) L_1$ and L_2 , the following system of conditions is carried out for them

$$Det M_{k \times k} \cap (A)_j \subseteq L_1 \cup L_2,$$

$$L_1 \setminus L_2 \cup L_2 \setminus L_1 \subseteq PerM_{k \times k} \cap (A)_i$$
.

We must add that the union of linear combinations $L_1 \cup L_2$ of rank columns (rows) is a linear combination of rank columns (rows).

In the case when elements of the matrix A are chosen from B_2 the above mentioned formulas are simplified and written down as

$$L_1 \setminus L_2 \cup L_2 \setminus L_1 \subseteq (A)_j \subseteq L_1 \cup L_2$$

for any column $(A)_j$ (or row correspondingly) of this Boolean matrix A and some linear combinations L_1 and L_2 of rank columns (rows).

The following statement is especially suitable for evaluation of a factor-rank (rank of factorization or Boolean rank or Schein rank) of Boolean matrix.

Theorem 4. A minor rank function rank gives an exact lower boundary for values of a term-rank function $rank_T$ and factor-rank function $rank_F$, that is

$$rankA \leq rank_T A$$
,

$$rankA \leq rank_F A$$

hold for all matrices A with elements from arbitrary Boolean algebra.

The Bibliography

- [1.] Rutherford, D. E. Inverses of Boolean matrices. Proc. Glasgow Math. Assoc. 1963, Vol. 6, N1. P. 49-53.
- [2.] Golan, J. S. Semirings and their Applications. Dordrecht: Kluwer Academic Publishers. 1999. xi, 381
- [3.] Poplin, P. L.; Hartwig, R. E. Determinantal identities over commutative semirings. Linear Algebra Appl. 387, 99-132, 2004.
- [4.] Chesley, D. S., Bevis, J. H. Determinants for matrices over lattices. Proc. Roy. Soc. Edinburgh, A 68, 2, p. 138 144, 1969,
- [5.] Poplavski, V. B. Orientation and permanent decomposition of Boolean matrices. Abstracts The 9th Asian Logic Conference 16-19.08.2005. Novosibirsk, Russia. Novosibirsk State University, Sobolev Institute of Mathematics. P.117-119.

Anticommutative Groebner Bases in Algebraic Geometry. Analysis of Geometrical Theorems in Coordinate-Free Form.

Irina J. Tchoupaeva

 $Moscow,\ MSU,\ mech-math \\ E-mail:\ ladyirina@shade.msu.ru,\ ladyirina@mail.ru$

In this paper we analyse the property of the theorem considered as a set of hypothesys and conclusions. The minimum number $k(\mathbb{T})$ so that $\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_k\}$ let us in coordinate method to reduce the complexity of calculation when we prove the theorem automatically. And the maximum number $d(\mathbb{T})$ so that $\{\mathbb{H}, \dots, \mathbb{H}_s, \mathbb{S}p_{d(\mathbb{T})}\} \Rightarrow \{\mathbb{C}onc\}$ let us to find the space in which the theorem is generally true.

Introduction

Automation of the prooving geometrical theorems was a big problem for the mathematicions during long time. And one of the main work in this domain is Chou's book [2] with the collection of two dimentional theorems formulated in the terms of constructive type statements. All these theorems was proved mechanically. And for these theorems were found nondegenerate conditions. Chou works with two dimentional theorems and formulates statements in coordinate form. So he can apply a Wu's or Gröbner bases methods for the prooving. He introduce the statements of the constructive type and formulate theorems in these terms. Wu's method and method of commutative Gröbner bases was described in detail in the papers [2,4,12].

Wang [12] also works with automated theorem prooving by using Gröbner bases method and introduce a coordinate free approach. He notes that there is possible to use Gröbner bases in Grassman algebras to prove some theorems.

In our previous papers [8,9] we classify statements which can be formulated in the terms of Grassman algebra polynomials. Also we found what kind of polynomials in Grassman algebra may be treated as geometrical statements.

In this paper we consider theorems of elementary geometry. And analyse them as a set of hypotheses and conclusions. We found a property of the set of theorem's hypothesys, named dimension of the theorem. Afther that we analyse where we may use this property. We consider the theorems in coordinate free form, so we have no coordinate system, but we can use this property to optimise the choosing of coordinate system by minimising the number of variables in the polynomial system. Why we return to the coordinate form? Because in coordinate method we do not have the limitation for class of the geometrical theorems such that we have in coordinate free method. And we can consider more wide set of the theorems.

In coordinate case we have to make the preliminary analysis by hand when we choose the good coordinate system to minimize number of variables and equations. But using information about dimension of the task we may automate the preliminary analysis.

Usually when the authors consider the theorems in coordinate form they fix the dimension of the space in which they prove the theorem. Some theorem may be not proved in the n-dimensional space but the projection of this theorem to the m-dimensional space with m < n may became proved. We also can make this analysis. We can apply this results not only for coordinate free method, but for the coordinate method too.

Coordinate free representation

Primarily, let us describe objects and tasks which will be regarded. We consider geometrical theorems in m - dimensional space \mathbb{R}^m , where $m \in \mathbb{N}_0$.

Let $\{A_1, A_2, A_3, \dots, A_n\} \in \mathbb{R}^m$ be points of the task. We treat these points as vectors drawn from origin 0.

Data of a theorem contains a finite number of points A_1, \ldots, A_n and finite number of k_1, \ldots, k_l dimensional subspaces of \mathbb{R}^m , $k_1, \ldots, k_l \leq m$ and their properties. Let be $m \geq (n-1)$, because in general case n points define a (n-1) - space, and if we consider a space with m < (n-1), we have some limitations for initial independent points.

Then, geometrically, the outer product of two vectors A and B is the bivector corresponding to the parallelogram obtained by sweeping vector A along vector B. The parallelogram obtained by sweeping B along A differs from the parallelogram obtained by sweeping A along B only in orientation. Let us consider the algebra generated by points $A_1, A_2, A_3, \ldots, A_n$ with an outer product $A \wedge B$, which is associative and anticommutative: $A \wedge B = -B \wedge A$. This algebra is called *Grassman algebra*.

The following statements of algebraic geometry may be written as polynomials in Grassman algebra:

1. (k+2) points $A_1, \ldots, A_{k+2}, k+2 \le n \le m+1$ belong to the same k - dimensional subspace \mathbb{R}^k of \mathbb{R}^m : $(A_1 - A_{k+2}) \land \cdots \land (A_{k+1} - A_{k+2}) = 0$ or in other words point $A_{i_{k+2}}$ belongs to k - subspace defined by the points $A_{i_1}, \ldots, A_{i_{k+1}}$:

$$(A_{i_1} - A_{i_{k+2}}) \wedge \cdots \wedge (A_{i_{k+1}} - A_{i_{k+2}}) = 0$$

where $\{i_1, ..., i_{k+2}\} = \{1, ..., k+2\}$ as non-ordered sets

2. two k - dimensional spaces are parallel $S_1 || S_2 \subset \mathbb{R}^m$, this means that $\forall (k+1)$ points $A_1, \ldots, A_{k+1} \in S_1$ and any 2 points $B_1, B_2 \in S_2$:

$$(A_1 - A_{k+1}) \wedge \cdots \wedge (A_k - A_{k+1}) \wedge (B_1 - B_2) = 0$$

- 3. (k+2) points $A_1, \ldots, A_{k+2}, k+2 \le n \le m+1$ belong to the same k dimensional subspace \mathbb{R}^k of \mathbb{R}^m and the proportion is known: $\alpha_1(A_1-A_{k+2})+\cdots+\alpha_{k+1}(A_{k+1}-A_{k+2})=0$, where $\alpha_1,\ldots,\alpha_{k+1}\in\mathbb{R}$
- 4. as the generalization of the previous expressions, that there is a linear dependency among a finite number of k-vectors

$$\sum_{(i_1,\dots,i_k)\in\mathbb{R}^k} \alpha_{(i_1,\dots,i_k)} A_{i_1} \wedge \dots \wedge A_{i_k} = 0$$

All homogeneous polynomials may be treated as some statements of algebraic geometry.

The system of polynomials corresponding to the hypotheses of the theorem are considered as generators of an ideal in Grassman algebra.

Let the theorem consist of a number of hypotheses $\mathbb{H}_1, \dots, \mathbb{H}_s$ and a conclusion $\mathbb{C}onc$. Then, these geometrical statements correspond to polynomials $h_1, \dots, h_s \in Gr$ and $conc \in Gr$.

Grassman algebra is generated by points of our theorem A_1, \ldots, A_n ,

 $Gr = Gr(A_1, \dots, A_n)$. Geometrical statements (hypotheses of the theorem) are formulated as polynomials in Gr:

$$h_1(A_1, \dots, A_n) = 0$$

$$\vdots$$

$$h_s(A_1, \dots, A_n) = 0$$

Let be $G = \{g_1, \dots, g_q\} \subset I \lhd Gr$ the finite Gröbner basis of the ideal I. We can find it using the algorith described in [8,9].

Definition. Let $\{h_1, \ldots, h_s\} \subset Gr$ be a set of polynomials corresponding to the hypotheses of the theorem and $conc \in Gr$ be a polynomial corresponding to the conclusion of the theorem. We say, that theorem is generally true, if for each partial solution (A_1^0, \ldots, A_n^0) of the system $h_1 = 0, \ldots, h_s = 0$, we have

$$conc(A_1^0, \dots, A_n^0) = 0.$$

Definition. Let $I \triangleleft Gr$ be an ideal in Grassamn algebra. The *pseudo radical* of the ideal be $\sqrt{I}^{ps} = \{f \in Gr \mid \exists \ m \in \mathbb{N} \ f^m \not\equiv 0, \ f^m \in I\}.$

```
Note that \sqrt{I}^{ps} = \{ f \in Gr \mid \exists \ 1 \le m \le n \ f^m \not\equiv 0, \ f^m \in I \}.
```

The property $f \in I$ is equivalent to zero reducibility of this polynomial f by G:

$$\mathbf{NForm}(f,G) = 0 \text{ in } Gr.$$

The following algorithm let us to check if $f \in \sqrt{I}^{ps}$ in Gr:

Algorithm:

```
Input: I \lhd Gr, f
Output: the answer if f \in \sqrt{I}^{ps}
begin

Calculate Gröbner basis of I: G

for k=1 to n do

fk=f^k;

if (fk\not\equiv 0) then

NF=\mathbf{Nform}\;(fk,G);

if (NF=0) then

return "f \in \sqrt{I}^{ps}"

endif

else

return "f \not\in \sqrt{I}^{ps}"

endif

endfor

return "f \not\in \sqrt{I}^{ps}"
```

Thus, the idea is to take the theorem, reformulate it as a set of polynomials in Gr, calculate Gröbner basis of the ideal generated by these polynomials and verify whether $f \in \sqrt{I}$ in Gr or not.

Analysis of the theorem So, we have a comprehensive space \mathbb{R}^m , and the task with its n points $A_1, \ldots, A_n \in \mathbb{R}^m$ and its.

Let $\mathbb{S}p_k$ the set of statements defined that all points of the task belong to the same k-dimensional space (for each set of points $\{A_{i_1},\ldots,A_{i_{k+2}}\}\subset\{A_1,\ldots,A_n\}$ can be constructed a polynomial by using previous rules). We obtain C_n^{k+2} polynomials.

For the fixed $1 \le k \le m$ there are two kinds of the theorems, which we consider:

- 1. hypotheses of the theorem $\bar{\mathbb{H}} = \{\mathbb{H}_1, \dots, \mathbb{H}_s\}$ imply the statement, that all points of this theorem lie in the same k dimensional subspace \mathbb{R}^k of \mathbb{R}^m
- 2. all other theorem

For the first class of theorem, we obtain the following:

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s\}\Rightarrow\{\mathbb{S}p_k\}$$

It would be noticed, that if we have n point A_1, \ldots, A_n in the task, then for any compatible set of hypotheses we have

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s\}\Rightarrow \{\mathbb{S}p_{n-1}\}$$

because any of n points belong to the same (n-1) - dimensional space.

And we have that for $\forall \mathbb{T} \exists 1 \leq k(\mathbb{T}) \leq m$ so that $\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_{k(\mathbb{T})}\}.$

From the other hand, for any $1 \le k \le m$ we have

$$\{\mathbb{S}p_k\} \Rightarrow \{\mathbb{S}p_{k+1}\}$$

and it can be verified by using Gröbner bases method too.

And we have the following:

$$(\{\mathbb{H}_1,\ldots,\mathbb{H}_s\}\Rightarrow\mathbb{C}onc)\Rightarrow(\{\mathbb{H}_1,\ldots,\mathbb{H}_s,\mathbb{S}p_k\}\Rightarrow\mathbb{C}onc)$$

It means that if theorem is generally true in \mathbb{R}^m , then the projection of this theorem to the subspace $\mathbb{R}^k \subset \mathbb{R}^m$ is also generally true in \mathbb{R}^k , $\forall 1 \leq k \leq m$.

So, we obtain, if $\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_k\}$ and the theorem is generally true, then this theorem is generally true in any subspaces Sp_k for $0 \le k \le n-1$.

Definition. Dimension of the theorem is a minimal number $k: 1 \leq k \leq m$ so that $\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_k\}$;

For finding the dimension $k(\mathbb{T})$ of the theorems we can use the following algorithm:

Algorithm:

```
\begin{split} \textbf{Input:} \ \mathbb{T} &= \{\mathbb{H}_1, \dots, \mathbb{H}_s; \mathbb{C}onc\} \\ \textbf{Output:} \ k(\mathbb{T}) \\ \textbf{begin} \\ & \textbf{for} \quad i = 0 \quad \textbf{to} \quad n-1 \quad \textbf{do} \\ & \textbf{Prove} \ (\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_i\}) \\ & \textbf{if} \ (TRUE) \ \textbf{then} \\ & \textbf{return} \ i \\ & \textbf{endif} \\ & \textbf{endfor} \\ \textbf{end} \end{aligned}
```

We can see, that dimension of the theorem is the property of the hypotheses. And in the other words, dimension of the theorem is the minimum number $1 \le k(\mathbb{T}) \le (n-1)$:

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s\}\Rightarrow\{\mathbb{S}p_{k(\mathbb{T})}\}$$

This number $k(\mathbb{T})$ exists for each theorem, because for any set of hypotheses $\mathbb{H}'_1, \ldots, \mathbb{H}'_{s'}$ the theorem $\mathbb{T}' = \{\mathbb{H}'_1, \ldots, \mathbb{H}'_{s'}; \mathbb{S}p_{n-1}\}$ will be generally true.

But how the information about $k(\mathbb{T})$ may help us and where we can use it? As the reader can see, if we use coordinate method in general case (without preliminary analysis of the coordinate system) we have $n \cdot m$ variables x_i^j :

$$A_1(x_1^1, \dots, x_m^1)$$

$$\vdots$$

$$A_n(x_1^n, \dots, x_m^n)$$

But, if $\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{S}p_{k(\mathbb{T})}\}$, then there is no reason to take a m-dimensional coordinate system. It is enought to take only $k(\mathbb{T})$ -dimensional coordinate system. This notice helps us to reduce the number of variable in polynomial system, therefor the complexity of calculations also reduces.

In coordinate free method the complexity of the calculation is not so big as in coordinate method. But in coordinate method we have

- 1. We don't make a preliminary analysis by hand, but work with big number of variable.
- 2. We make the preliminary analysis by hand, but we get the limitation for the dimension of the space, because a human usually operate with the space of small dimension (2D -theorem or 3D -theorem).

Thus, the finding of the $k(\mathbb{T})$ helps us to reduce the complexity of the calculation in the coordiname method.

Next, we can consider the common property of hypotheses and conclusion of the theorem as the maximum number $1 \leq d(\mathbb{T}) \leq m$ so that

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s,\mathbb{S}p_{d(\mathbb{T})}\}\Rightarrow \{\mathbb{C}onc\}.$$

Note, that if

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s,\mathbb{S}p_d\}\Rightarrow \{\mathbb{C}onc\},\$$

then

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s,\mathbb{S}p_{d-1}\}\Rightarrow \{\mathbb{C}onc\}.$$

Using information about dimension of the theorem, we can make some remarks about finding additional conditions for the data, if the theorem is not generally true in \mathbb{R}^m . For example, if we find $1 \leq k(\mathbb{T}) \leq m$:

$$\{\mathbb{H}_1,\ldots,\mathbb{H}_s\}\Rightarrow \{\mathbb{S}p_{k(\mathbb{T})}\}$$

but

$$\{\mathbb{H}, \dots, \mathbb{H}_s\} \not\Rightarrow \{\mathbb{C}onc\}$$

we can try to find the maximum dimension $d(\mathbb{T})$ of the space in which this theorem is generally true :

$$\{\mathbb{H}, \dots, \mathbb{H}_s, \mathbb{S}p_{d(\mathbb{T})}\} \Rightarrow \{\mathbb{C}onc\}.$$

Additional conditions $\mathbb{S}p_{d(\mathbb{T})}$ will be restrictions for the A_1, \ldots, A_n , if $1 \leq d(\mathbb{T}) < k(\mathbb{T})$. This number $d(\mathbb{T})$ exists for each theorem, because for any set of hypotheses $\mathbb{H}'_1, \ldots, \mathbb{H}'_{s'}$ and any conclusion $\mathbb{C}onc'$ the theorem $\mathbb{T}' = \{\mathbb{H}'_1, \ldots, \mathbb{H}'_{s'}, \mathbb{S}p_0; \mathbb{C}onc'\}$ will be generally true.

So, the algorithm for finding $d(\mathbb{T})$ is the following:

Algorithm:

```
Input: \mathbb{T} = \{\mathbb{H}_1, \dots, \mathbb{H}_s; \mathbb{C}onc\}, k(\mathbb{T})

Output: d(\mathbb{T})

begin

For i = k(\mathbb{T}) to 1 do

Prove (\{\mathbb{H}_1, \dots, \mathbb{H}_s, \mathbb{S}p_i\} \Rightarrow \{\mathbb{C}onc\})

if (TRUE) then

return i

endif

endfor
```

Example. For example, the Gauss line theorem [2] the hypotheses imply that all points belong to the same plane (it is easy to check). So, in this theorem we have $\{\mathbb{H}_1, \ldots, \mathbb{H}_7\}$ hypotheses, and $(\{\mathbb{H}_1, \ldots, \mathbb{H}_7\} \Rightarrow \{\mathbb{S}p_2\})$. Thus $k(\mathbb{T}) = 2$.

But in the Pappus theorem [2] hypotheses of 9 points do not imply this property $\mathbb{S}p_2$. Moreover, this theorem is not generally true in the spaces \mathbb{R}^f with $9 \ge f \ge 3$ and $d(\mathbb{T}) = 2$.

If we use the coordinate method, we add the properties of the q - dimension space, when we introduce the coordinate system (e_1, \ldots, e_q) . This is equivalent to adding properties $\mathbb{S}p_q$ to the hypotheses.

When we consider not only hypothesys of the theorem but also the conclusion of the theorem in \mathbb{R}^m we have two possibilities:

1.
$$\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \Rightarrow \{\mathbb{C}onc\}$$

2.
$$\{\mathbb{H}_1, \dots, \mathbb{H}_s\} \not\Rightarrow \{\mathbb{C}onc\}$$

But in the second case we can find $d(\mathbb{T})$ so that $\{\mathbb{H}_1, \dots, \mathbb{H}_s, \mathbb{S}p_d\} \Rightarrow \{\mathbb{C}onc\}$. And find in which space the theorem became generally true.

Conclusions

• coordinate free method let us to reduce the complexity of the calculation during the authomated geometrical theorem prooving, but the class of the theorem we can consider in coordinate free method is smaller that in coordinate method;

- the finding the dimensional of the theorem $k(\mathbb{T})$ let us to automate the preliminary analysis for choosing the coordinate system and reduce the number of the variable in general case. There for we have the reduction of the complexity of calculations;
- the finding of the maximum dimension of the space in which the theorem is generally true let us to analyse when the theorem is true, if the theorem is not true in general

Acknowledgments. I am grateful to my advisor E.V. Pankratiev for suggestions while preparing this paper and for many valuable remarks.

The Bibliography

- 1. B. Buchberger, Gröbner bases: an algorithmic method in polynomial ideal theory, in N.K.Bose (ed.), *Recent trends in multidimensional system theory*, Reidel, 1985 *Mathematics and Its Applications*, D. Reidel, Dordrecht, 1987.
- 2. S.C. Chou, Mechanical Geometry Theorem Proving *Mathematics and Its Applications*, D. Reidel, Dordrecht, 1987.
 - 3. D.Cox, J.Little, D. O'Shea, Using Algebraic Geometry, Springer, New-York, 1998
- 4. D.Cox, J.Little, Ideals, Varieties, and Algorithms An Introduction to Computational Algebraic Geometry and Commutative Algebra, second edition, Springer, New-York, 1998
- 5. D. Eisenbud, I. Peeva, B. Sturmfels, Noncommutative Gröbner Bases for Commutative Ideals. *To appear in Proc. Am. Math. Soc.*
 - 6. T. Mora, Gröbner bases in noncommutative algebras, Proc. ISSAC 88, LNCS 358, 1989, pp. 150-161
- 7. T.Mora: An Introduction to Commutative and Noncommutative Gröbner Bases, *Theoretical Computer Science* vol.134 p131-173, 1994.
- 8. I. J. Tchoupaeva: Automated proving and analysis of geometric theorems in coordinate-free form by using the anticommutative Grobner basis method. *Journal of Mathematical Sciences, Springer US, vol. 135, no. 5, pp. 3409-3419* (Translated from Fundamentalnaya i Prikladnaya Matematika, Vol. 9, No. 3, pp. 213-228, 2003).
- 9. I.J.Tchoupaeva: Analysis of geometrical theorems in coordinate-free form by using anticommutative Grobner bases. LNAI 2930 sub series of LNCS, Springer-Verlag, pp.178-193
- 10. V.Ufnarovski: Introduction to Noncommutative Gröbner Bases Theory, in Gröbner Bases and Applications, B.Buchberger and F.Winkler (eds), Proc. London Math. Soc. vol.251, 1998.
- 11. V.W.Vasconcelos, Computational Methods in Commutative Algebra and Algebraic Geometry, Springer Verlag, 1998 Verschelde 1999
- 12. D. Wang, Gröbner Bases Applied to Geometric Theorem Proving and Discovering, *Gröbner Bases and Applications*, Eds. B. Buchberger and F. Winkler, Cambridge Univ. Press, 1998, pp. 281–302.

Схема асимметричного шифрования, основанная на отечественных криптографических примитивах

Аносов В. Д., Нестеренко А. Ю.

В работе [1] авторами была предложена общая схема асимметричного шифрования с открытым ключом, которая может быть реализована в клиент-серверных приложениях. Основной особенностью данной схемы являлась возможность аутентификации клиента, отправившего сообщение.

В настоящей работе мы рассматриваем одну из возможных модификаций предложенной в [1] схемы, основанную на отечественных криптографических примитивах, а именно алгоритме блочного шифрования ГОСТ 28147-89 [2] и схеме электронной цифровой подписи ГОСТ Р 34.10-2001 [4].

Параметры схемы шифрования

Для описания предлагаемой нами модификации схемы шифрования с открытым ключом, необходимо определить следующие математические объекты и криптографические преобразования

ullet Группу точек эллиптической кривой \mathcal{E} , заваемой сравнением

$$y^2 \equiv x^3 + ax + b \pmod{p},$$

где $p>2^{511}$ — простое число, и $4a^3+27b^2\neq 0\pmod p$. Порядок данной группы мы будем обозначать символом m.

- Точку P эллиптической кривой \mathcal{E} , которая порождает циклическую подгруппу простого порядка q, для которого выполнено условия
 - 1. $m = qn, 1 \le n \le 4,$
 - $2. \ 2^{511} < q < 2^{512}.$
- Функцию v, отображающую точки эллиптической кривой $\mathcal E$ в два целых числа $k_1,k_2,\ 0< k_1,k_2<2^{256}$. Функция v действует следующим образом

$$v(P) = (k_1, k_2), \quad P \in \mathcal{E}, \quad k_1, k_2 \in \mathbb{N},$$

$$k_1 \equiv p_x \pmod{2^{256}}, \quad k_2 \equiv 2^{-256} (p_x - k_1) \pmod{2^{256}},$$

где p_x — x-координата точки P.

- Алгоритм E блочного шифрования сообщений ГОСТ 28147-89 [2], использующий для шифрования и расшифрования ключи длиной 256 бит. Мы будем использовать обозначение E(k,M) для обозначения зашифрования сообщения M на ключе k.
- Схему электронной цифровой подписи ГОСТ Р 34.10-2001 [4] и функцию хеширования ГОСТР Р 34.11-94 [3]. Отметим, что длина инициализационного вектора данной функции равна 256 бит.
- Функцию || конкатенации двух двоичных векторов произвольной длины.

Отметим, что все перечисленные выше параметры и алгоритмы долны быть известны всем участникам схемы шифрования.

Определим ключевую систему рассматриваемой нами схемы шифрования. Сервер, являющийся получателем сообщения, обладает следующими ключами

- 1. Секретным ключом x двоичным вектором длины s.
- 2. Открытым ключом Q точкой эллиптической кривой $\mathcal{E},$ удовлетворяющей групповому равенству Q=xP.

Клиент, желающий отправить сообщение серверу, должен обладать секретным x_k и открытым Q_k ключами ЭЦП, используемыми для выработки и проверки электронной цифровой подписи.

Алгоритмы зашифрования и расшифрования сообщений

Для зашифрования произвольного сообщения M клиент выполняет следующую последовательность действий.

- Шаг 1. Клиент вырабатывает псевдо-случайное целое число k, удовлетворяющее неравенствам 0 < k < q.
- Шаг 2. Клиент вычисляет две точки U и T эллиптической кривой \mathcal{E} , удовлетворяющие групповым равенствам

$$U = kP$$
, $T = kQ$,

где P точка, порождающая подгруппу простого порядка q, а Q — открытый ключ сервера, получателя сообщения M.

- Шаг 3. С помощью функции v клиент вырабатывает два сеансовых ключа k_1 и k_2 , удовлетворяющих равенству $v(T)=(k_1,k_2)$, и представляет их в виде двоичных векторов длины 256 бит каждый.
- Шаг 4. С помощью алгоритма блочного шифрования E клиент зашифровывает сообщение M на сеансовом ключе k_1 и вырабатывает шифртекст e, удовлетворяющий равенству $e = E(k_1, M)$.
- Шаг 5. Используя x_k секретный ключ ЭЦП, клиент вырабатывает цифровую подпись w под шифртекстом e, используя в качестве начального вектора h_0 функции хэширования значение k_2 , выработанное на 3-м шаге данного алгоритма.
- Шаг 6. С использованием функции конкатенации клиент формирует двоичную последовательность $\xi = u_x ||u_y||w||e$, где u_x и u_y соответсвенно, x- и y-координаты точки U, выработанной на втором шаге данного алгоритма.

Двоичная последовательность ξ является шифртекстом, соответствующим открытому тексту M, и направляется серверу.

Для расшифрования сообщения ξ сервер выполняет следующую последовательность действий.

- Шаг 1. Из полученного сообщения ξ сервер выделяет значения u_x, u_y, w, e и проверяет, что пара чисел (u_x, v_x) корректно задает точку U эллиптической кривой \mathcal{E} .
- Шаг 2. Используя свой секретный ключ x, сервер вычисляет точку T, удовлетворяющую равенству T=xU.
- Шаг 3. С помощью функции v сервер вырабатывает сеансовые ключи k_1 и k_2 , удовлетворяющие равенству $v(T) = (k_1, k_2)$.
- Шаг 4. Сервер проверяет подлинность цифровой подписи e под шифртекстом e, используя в качестве начального вектора h_0 функции хэширования значение k_2 , выработанное на 3-м шаге данного алгоритма.
- Шаг 5. Если цифровая подпись верна, то с помощью ключа k_1 сервер расшифровывает сообщение M.

Заключение

Предложенная нами схема асимметричного шифрования позволяет клиенту передавать зашифрованные сообщения по открытым каналам связи без предварительного распределения секретных ключей. Используемые в схеме отечественные криптографические примитивы позволяют обеспечить высокий уровень засекречивания информации. Кроме этого, данная схема обладает следующими особенностями:

- Нарушитель не может подменить или модифицировать передаваемую информацию, поскольку ее целостность подтверждается цифровой подписью.
- Использование ЭЦП позволяет обеспечить аутентичность клиента, отправившего сообщение.
- При шифровании одного и того же открытого текста в канал связи передается разный шифртекст
- Размер шифртекста превышает размер открытого текста на малую фиксированную величину.

The Bibliography

- 1. Аносов В.Д., Нестеренко А.Ю. Схема асимметричного шифрования с возможностью аутентификации, препринт, 2006.
- 2. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. 1989.
- 3. ГОСТ Р34.11-94 Информационная технология. Криптографическая защита информации. Функция хэширования. 1994.
- 4. ГОСТ Р 34.10-2001 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. 2001

О замкнутых классах автоматных функций

Бабин Д. Н.

dbabin@mech.math.msu.su

каф. МаТИС, мех-мат ф-т МГУ, им. Ломоносова

Рассматривается алгебра автоматов с операцией суперпозиции. Исследована вложенная последовательность бесконечнопорожденных классов автоматов, имеющих бесконечную высоту в своем непосредственном надклассе.

Пусть $E_2 = \{0,1\}$ $E_2^n = E_2 \times E_2 \times ... \times E_2$ - декартово произведение n копий E_2 , через $\left(E_2^n\right)^k = \{a(1)a(2)...a(k) \,|\, a(1),a(2),...,a(k) \in E_2^n\}$ обозначается множество слов длины k в алфавите E_2^n , через $\left(E_2^n\right)^* = \bigcup_{k=1}^{\infty} \left(E_2^n\right)^k$ — множество всех конечных слов в алфавите E_2^n , а через $\left(E_2^n\right)^\infty = \{a(1)a(2)... \,|\, a(j) \in E_2^n, \, j=1,2,...\}$, — множество бесконечных слов (сверхслов) в алфавите E_2^n . Функции вида $g: E_2^n \to E_2$ называются булевыми, а функции $g: E_2^n \to E_2^m$ булевыми векторфункциями.

Будем рассматривать автоматные функции на бесконечных последовательностях из нулей и единиц (сверхсловах)

$$f: (E_2^n)^{\infty} \to (E_2^m)^{\infty} \tag{1}$$

как функции, задаваемые рекуррентными соотношениями (2)

$$\begin{cases} q(1) = q_1, \\ q(t+1) = \varphi(q(t), a_1(t), ..., a_n(t)), \\ f_j(t) = \psi_j(q(t), a_1(t), ..., a_n(t)), j = 1, ..., m \end{cases}$$
(2)

где $q\in Q_f=\{q_1,q_1,...q_s\}$ конечное множество состояний функции f . Такие функции называются автоматными функциями (a-функциями) с n входами и m выходами, параметр q при этом называется состоянием a-функции f; q_1 - начальным состоянием, вектор-буквы $a=(a_1,...,a_n)$ и $b=(b_1,...,b_m)$ - входной и выходной буквами, а сверхслова a(1)a(2) ... и b(1)b(2) ... - входным и выходным сверхсловами, соответственно. Шестерка $A=\left(E_2^n,Q_f,E_2^m,\varphi,\psi,q_1\right)$ называется автоматом, порождающим a-функцию f . Функции φ и ψ называются функцией переходов и выходной функцией автомата, соответственно. Заметим, что разные автоматы могут порождать одну и ту же a-функцию.

Введем операции суперпозиции, позволяющие получать новые автоматные функции из уже имеющихся [1]. Здесь η и $\mathcal E$ операции перестановки входов, ω операция отождествления входов, δ операция удаления фиктивного входа, \ast операция подстановки выходов одной функции на входы другой. Для функций $f: (E_2^n)^{\infty} \to (E_2^m)^{\infty}$, $g: (E_2^r)^{\infty} \to (E_2^{k+s})^{\infty}$, $(f*g): (E_2^{n-s+r})^{\infty} \to (E_2^{m+k})^{\infty}$ операции суперпозиции задаются уравнениями (3).

$$\begin{cases} (\eta f)(x_{1}, x_{2}, ...x_{n}) = f(x_{2}, x_{3}, ...x_{n}, x_{1}) \\ (\varepsilon f)(x_{1}, x_{2}, ...x_{n}) = f(x_{2}, x_{1}, x_{3}, ...x_{n}) \\ (\omega f)(x_{1}, x_{2}, ...x_{n-1}) = f(x_{1}, x_{1}, x_{2}, ...x_{n-1}) \end{cases}$$

$$(\delta f)(x_{1}, x_{2}, ...x_{n+1}) = f(x_{2}, x_{3}, ...x_{n+1}) \\ (f * g)(x_{1}, x_{2}, ...x_{n-k+r}) = \\ = (g_{1}(x_{1}, ...x_{r}), ..., g_{k}(x_{1}, ...x_{r}), f(g_{k+1}(x_{1}, ...x_{r}), ..., g_{k+s}(x_{1}, ...x_{r}), x_{k+s+1}, ...x_{k+n})) \end{cases}$$

Будем считать, что $Q_f\equiv Q_{nf}\equiv Q_{\varepsilon f}\equiv Q_{\omega f}\equiv Q_{\delta f}$, $Q_{f*\varrho}\equiv Q_f\times Q_\varrho$

Предполагается, что тождественная одноместная булева функция $h(x) \equiv x$ и функция селектор $w(x,y) \equiv x$ всегда имеются в нашем распоряжении. Из $h(x) \equiv x$ отождествлением входов может быть получена функция $H_2: E_2 \to E_2^2, H_2(x) = (x,x)$,

подстановка на которую, позволяет получить раздвоения выхода. Подстановка на функцию $w(x, y) \equiv x$, позволяет удалить выход функции.

Заметим, что в определении операции подстановки число S выходов a-функции g, подставляемых на входы a-функции f, без ограничения общности может быть принято за 1. А также можно считать, что все выходы a-функции f подставляются на все входы a-функции g.

Класс всех a-функций обозначим через ${\bf A}$, множество a-функций с одним состоянием через ${\bf A}_1$, назовем одноместными автоматные функции (1) при n=1, двухместными – при n=2, и.т.д. Множество одноместных функций обозначим через ${\bf A}^1$, двухместных – через ${\bf A}^2$, . . . n местных – через ${\bf A}^n$. Для $\nu \subseteq {\bf A}$ обозначим через $\nu_1 \subseteq {\bf A}_1 \cap \nu$, $\nu_2 \subseteq {\bf A}_1 \cap \nu$.

Пусть $\nu \subset \mathbf{A}$ обозначим через $[\nu]$ множество a-функций, получающихся из функций множества ν применением конечного числа операций суперпозиции, множество $[\nu]$ называется замыканием множества ν относительно суперпозиции, $\nu = [\nu]$ называется замкнутым классом, ν называется полным относительно суперпозиции, если $[\nu] = \mathbf{A}$. Из того, что

$$Q_f \equiv Q_{\eta f} \equiv Q_{\varepsilon f} \equiv Q_{\omega f} \equiv Q_{\delta f} \,,\, Q_{f*g} \equiv Q_f \times Q_g$$

следует, что \mathbf{A}_1 замкнутый класс. Автоматную функцию из \mathbf{A}_1 мы будем отождествлять с булевой вектор-функцией $h:E_2^n\to E_2^m$, доопреденной до автоматной функции $h:\left(E_2^n\right)^\infty\to \left(E_2^m\right)^\infty$ соотношением h(a(1)a(2)...)=h(a(1))h(a(2))....

Известно [2], что множество всех булевых функций получается суперпозициями булевых функций дизъюнкция V, конъюнкция &, отрицание \neg , и даже суперпозициями одной функции. Заметим, что булевы вектор-функции получаются суперпозициями из булевых функций и функции H_2 . Там, где это не приводит к противоречию, мы будет называть булевы вектор-функции булевыми функциями. Имеет место

Теорема 1. [3] Пусть $\mu = \{f_1, f_2, ..., f_r\} \subset \mathbf{A}$, конечная система автоматных функций тогда $[\mu] \neq \mathbf{A}$.

Рассмотрим множество автоматных функций ${\bf A_N}$, у которых число состояний (при каком-нибудь способе задания автоматом) не превосходит N . Выполнена цепочка включений

$$\mathbf{A}_1 \subset \mathbf{A}_2 \subset \ldots \subset \mathbf{A}_N \subset \mathbf{A}_{N+1} \subset \ldots \subset \mathbf{A}$$
.

Имеет место

Теорема 2[4]. Для любого N множество $\mathbf{A_{N}}$ порождается конечным числом автоматных функций.

Автоматные функции, выдающие своё состояние, называются автоматными функциями Медведева [3]. Для них будем считать, что $Q \subseteq E_2^m$, ϕ и ψ - булевы, а уравнения (2) имеют вид

$$\begin{cases} q(1) = (0,0,...,0) \\ q(t+1) = \varphi(q(t),a_1(t),...,a_n(t)), \\ b(t) \equiv q(t). \end{cases}$$
(4)

Определение. Пусть f и g — a.-функции с одинаковым числом входов и одинаковым числом выходов. Будем говорить, a.-функции g является копией a.- функции f если для натуральных $k,i \leq j$ и всякого входного сверхслова a = a(1)a(2)... для выходных сверхслов выполнено $g(\alpha)(j+kl) = f(\alpha)(i+kl),$ l = 0,1,2,...

Саму функцию f по отношению к копии g будем называть оригиналом. Отношение копирования является рефлексивным и транзитивным, и, вообще говоря, не симметричным.

На любой входной последовательности копия функции в моменты времени, образующие арифметическую прогрессию, выдает те же значения, что и оригинал.

Лемма 1. (о копировании.) [5] Пусть f а.-функция Медведева и g ее копия, тогда $f \in [\mathbf{A}_1, \mathbf{A}^1, g]$.

На основе леммы 1 получаются

Теорема 3. [5]
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \cup \mathbf{A}^1 \end{bmatrix}$$
.

Teopeма 4. [5]
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^2 \end{bmatrix}$$
.

Теорема 5. [6]
$$\mathbf{A}_{N} = \left[\mathbf{A}_{1} \bigcup \mathbf{A}_{N}^{1} \right].$$

Теорема 6. [6]
$$A_N = \begin{bmatrix} A_N^2 \end{bmatrix}$$
.

Более того, существует универсальная автоматная функция $f \in \mathbf{A}_{\mathrm{N}}^2$, такая что $\mathbf{A}_{\mathrm{N}} = [f]$ [6].

Фиксированная входная буква $a \in E_2^m$ задает подстановку множества состояний автомата $\left(\varphi\right)_a:Q \to Q$, где $\left(\varphi\right)_a(q)=\varphi(q,a)$. Последовательное применение подстановок, задаваемых буквами $a,b \in E_2^m$ это подстановка, задаваемая словом ab $\left(\varphi\right)_{ab}(q)=\left(\varphi\right)_b\left(\left(\varphi\right)_a(q)\right)$.

Множество подстановок состояний автомата образует полугруппу S_A с операцией $(\varphi)_b \bullet (\varphi)_a = (\varphi)_{ab}$, называемую полугруппой автомата A. Если при всех $a \in E_2^n$ выполнено $\varphi(q_i,a) \neq \varphi(q_j,a)$ при $q_i \neq q_j$, то каждая входная буква порождает биекцию или перестановку множества состояний. Такие автоматы называются групповыми или перестановочными.

В этом случае $S_A=G_A=\left\{(\varphi)_\alpha:Q o Q\,|\,\alpha\in\left(E_2^n\right)^*\right\}$ является группой автомата A , а также группой $G_f=G_A$ его автоматной функции f .

Пусть M некоторое множество групп, через $\langle M \rangle$ обозначим множество групп, получающихся из элементов M конечным числом применений операций взятия подгруппы, гомоморфного образа и расширения. Через G обозначим множество всех групп. Известно [7], что абелевы группы порождают множество разрешимых групп $R \subset G$ и выполнено $R \neq G$, $\langle R \rangle = R$, $\langle G \rangle = G$.

Известно, что ни R, ни G не порождаются конечным числом групп, а также G не порождается никаким множеством $R\bigcup \Gamma$, где $\Gamma \subset G$ конечно.

Групповой автомат назовем коммутативным, если его группа коммутативна, заданную этим автоматом функцию также назовем коммутативной. Обозначим множество групповых автоматных функций через ${\bf G}$, множество коммутативных а-функций обозначим через ${\bf K}$, множество групповых автоматных функций с разрешимыми группами через ${\bf R}$. Автомат называется автоматом с безусловными переходами, если $\varphi(q,a) \equiv \varphi(q)$

Обозначим множество групповых автоматных функций с безусловными переходами через ${\bf B}$. Если в уравнениях (2) булева функция ${\bf \phi}$ линейна, то соответствующий автомат называется автоматом с

линейными переходами, множество групповых функций, задаваемых такими автоматами, обозначим через L, а его замыкание через L = [L]. Очевидно, что

$$[B] = B \subset K, B \subset L, B_1 = L_1 = K_1 = R_1 = G_1 = A_1$$

Если автоматная функций f выразима через автоматные функции множества F операциями суперпозиции, то ее группа выразима через соответствующие операции над группами функций из F.

Имеет место

Лемма 2. [8] Если $f\subseteq \lceil F \rceil$, то $G_f\in \langle G_g\mid g\in F \rangle$.

Следствие 1.
$$[G] = G$$
, $[R] = R$, $[B] = B$.

По групповому автомату $A = (E_2^n, Q, E_2^m, \varphi, \psi, q_1)$, задающему функцию f построим автомат $A^+ = (E_2^{n+1}, Q, E_2^{m+1}, \varphi^+, \psi^+, q_1)$, имеющий дополнительный вход и дополнительный выход.

Для $a \in E_2^n$, $b \in E_2^m$ определим

$$\varphi^{+}(q,(a,0)) = \varphi(q,a) \qquad \qquad \psi^{+}(q,(a,0)) = (\psi(q,a),0)$$

$$\varphi^{+}(q,(a,1)) = (\varphi_{a})^{-1}(q) \qquad \qquad \psi^{+}(q,(a,1)) = (\psi((\varphi_{a})^{-1}(q),a),1)$$

А.-функцию автомата A^+ обозначим через f^+ . Очевидно, что $G_f = G_{f^+}$. А-функцию $f^s: \left(E_2^{sn}\right)^\infty \to \left(E_2^{sm}\right)^\infty$ определенную по формуле $f^s\left((a_1...a_s)(a_{s+1}...a_{2s})...\right) = (b_1...b_s)(b_{s+1}...b_{2s})...$, где $f\left(a_1...a_sa_{s+1}...a_{2s}...\right) = b_1...b_sb_{s+1}...b_{2s}...$, назовем S -той степенью функции f. По определению имеем $\left(f^s\right)^k = \left(f^k\right)^s = f^{sk}$, $f^1 = f$

Обозначим через $f^{(+\;-)}$ результат применения к групповой функции f последовательности операций

$$f(x_1,...,x_n) \to f^+(x_1,...,x_n,x_{n+1}) \to f^+(x_1,...,x_n,0101...) \to (f^+)^2((x_1(1)x_1(2)),...,(x_n(1)x_n(2)),(01)(01)...)$$

Обозначим через $F^+ = \left\{ f^+ \mid f \in F \right\}, \quad F^{(s)} = \left\{ f^s \mid f \in F \right\}, \quad F^{(+-)} = \left\{ f^{(+-)} \mid f \in F \right\}.$

Имеет место

Лемма 3. [9]

$$I$$
) Для $f \in \mathbf{G}$ выполнено $\left(f \in \left[\left\{ f_1, f_2, ..., f_k \right\} \right] \right) \Rightarrow \left(f^{(+-)} \in \left[\left\{ f_1^{(+-)}, f_2^{(+-)}, ..., f_k^{(+-)} \right\} \right] \right)$

2) для натурального S выполнено

$$(f \in [\{f_1, f_2, ..., f_k\}]) \Rightarrow (f^{(s)} \in [\{f_1^{(s)}, ..., f_k^{(s)}\}])$$

Следствие 2. Если $F \subseteq \mathbf{G}$ полна, то $F^{(+\;-)}$ полна, если $F \subseteq \mathbf{A}$ полна, то $F^{(s)}$ полна.

Следствие 3.
$$\mathbf{A}^{(s)} \subseteq \mathbf{A}, \ \mathbf{G}^{(s)} \subseteq \mathbf{G}, \ \mathbf{K}^{(s)} \subseteq \mathbf{K}, \ \mathbf{B}^{(s)} \subseteq \mathbf{B}, \ \mathbf{L}^{(s)} \subseteq \mathbf{L}$$

$$G^{+} \subseteq G, K^{+} \subseteq K, B^{(+-)} \subseteq B_{1}, L^{(+-)} \subseteq R$$

Из доказательства теоремы 1 прямо следует.

Теорема 7. Каждое из множеств $\mathbf{B} \subseteq \mathbf{L} \subseteq \mathbf{K} \subseteq \mathbf{R} \subseteq \mathbf{G} \subseteq \mathbf{A}$ не являются конечнопорожденным.

Лемма 4. [6]
$$\mathbf{R} = [\mathbf{K}] = [\mathbf{K}^1 \cup \mathbf{K}_1] \subset [\mathbf{G}^1 \cup \mathbf{G}_1] \subset \mathbf{G}$$

Скажем, что класс функций $\mathbf{C} \subseteq \mathbf{D}$ имеет высоту k в классе \mathbf{D} , если для некоторой системы функций $\{f_1, f_2, ..., f_k\} \subseteq \mathbf{D}$ выполнено $\left[\mathbf{C} \bigcup \{f_1, f_2, ..., f_k\}\right] = \mathbf{D}$, а для всякой системы функций меньшей мощности это не так.

Скажем, что класс функций $\mathbf{C} \subseteq \mathbf{D}$ имеет бесконечную высоту в классе \mathbf{D} , если для любой конечной системы функций $\{f_1, f_2, ..., f_k\} \subseteq \mathbf{D}$ выполнено $\left[\mathbf{C} \bigcup \{f_1, f_2, ..., f_k\}\right] \neq \mathbf{D}$

Имеет место

Теорема 8.

- 1) Класс $\mathbf{G} \subset \mathbf{A}$ имеет высоту 1 в \mathbf{A} , класс $\mathbf{B} \subset \mathbf{L}$ имеет высоту 1 в \mathbf{L} ,
- 2) в цепочке вложений $\mathbf{L} \subset \mathbf{R} \subset \left[\mathbf{G}^1 \bigcup \mathbf{G}_1\right] \subset \mathbf{G}$ каждый класс имеет бесконечную высоту в своем непосредственном надклассе.

Таким образом, последовательность замкнутых классов имеет вид $\mathbf{B} \overset{_1}{\subset} \mathbf{L} \overset{_{\sim}}{\subset} \mathbf{R} \overset{_{\sim}}{\subset} \begin{bmatrix} \mathbf{G}^1 \bigcup \mathbf{G}_1 \end{bmatrix} \overset{_{\sim}}{\subset} \mathbf{G} \overset{_1}{\subset} \mathbf{A}$

Автор выражает благодарность академику Кудрявцеву В. Б. за ценные указания.

Список литературы

- 1. Мальцев А. И., Итеративные алгебры и многообразие Поста, Алгебра и логика, 1966, т.5, N2, с.5-24.
- 2.Post E., Two-valued iterative systems of math. logik.Printston, 1941.
- 3. Автоматы, Сборник статей под редакцией Маккарти и Шеннона, ИЛ, Москва, 1956.
- 4. Кудрявцев В. Б., Функциональные системы, изд. МГУ, 1982.
- 5.Бабин Д. Н., О полноте двухместных о.д.-функций относительно суперпозиции, Дискретная математика, том 1, 1989, выпуск 4, с.86-91, Наука, Москва.
- 6. Бабин Д. Н., О суперпозициях в некоторых классах о.-д. функций, Логико-алгебраические конструкции, Тверь, 1992, с.17-22.
- 7. Каргополов М. И., Мерзляков Ю. И., Основы теории групп, М. Наука, 1982.
- 8. Теория автоматов, языков и полугрупп, Сборник статей под редакцией Арбиба М., М. ИЛ, 1974.
- 9. Бабин Д. Н., Выразимость автоматов при использовании вербальных операций, Материалы всесоюзного семинара по дискретной математике Изд. МГУ, 1986, с. 155-160.

k-Гиперпространство семиотико-хроматических гипертопографов как универсальная модель представления фактографических знаний¹

Баранович А. Е., barae@rambler.ru

Профессор кафедры математики и информатики Московского государственного лингвистического университета, Россия, 119992, г. Москва, ул. Остоженка 38

Ключевые слова — гиперпространство булево, гипертопографы семиотико-хроматические, интеллектуальные системы (антропоморфные), информации модель-универсум, *k*-топологизация множества (носителя), *k*-гиперпространство семиотико-хроматических гипертопографов, прагматика, представление и моделирование знаний, семиотика, семантика, сенсориум, теория графов (гиперграфов), теория решеток (шкал), топология дискретная с конечным носителем, ценность информации.

В рамках концепции синтеза модели-универсума информации формализованы и исследованы основные элементы абстрактной модели k-гиперпространства семиотико-хроматических ги́перто́погра́фов Γ S, являющегося результатом экспликации модели информационных форм существования материальных систем в модель представления фактографических знаний. Предлагаемая модель поглощает известные модели представления знаний и позволяет свести решение задач количественного анализа прагматических атрибутов информации к исследованию метрических и измеримых свойств k-гиперпространства Γ S

Подавляющее большинство используемых в настоящее время моделей представления знаний (фреймы, семантические сети / метасети, категориальные модели, продукционные, логические системы и т.п.) базируются на вербальной характеризации и интерпретации знаний в интеллектуальных системах (ИС). В свою очередь, нейронные сети и их производные (как модели функционирования нейронной структуры биологического мозга), в первоначальном, историческом, контексте ориентировались на моделирование отдельных компонент сенсориума ИС и распознавание, в частности, оптических образов (в том числе, с самообучением). Исследование процессов функционирования интегрированных интеллектуальных систем, в частности, в отношении проблемы измеримости семантико-аксиологических характеристик прагматической информации в ИС показало [1, 3], что одним из важных элементов ее решения является синтез универсальной модели всех взаимосвязанных информационных компонент ИС в динамике ее существования. Различные экспликации (проекции) модели-универсума должны обеспечить моделирование и сенсориума, и подсистем знаний и коммуникации, и пространство целей, и объектов внешней среды, их свойств и отношений.

В основу концепции синтеза модели-универсума информации (ИН) [2-3], характеризующей и отображающей основные свойства информационной организации и взаимодействия произвольных материальных систем (МС) объективной реальности (ОР) положен тезис о вторичности языка (как основы сигнально-семиотической коммуникации антропоморфных ИС) и его формальных моделей по отношению к первично-гомоморфным проективно-сенсорным и семантико-образным моделям информации в подсистемах знаний ИС.

Ключевые элементы предлагаемого подхода к синтезу модели ИН базируются на cosokynhocmu npuнципов, изложенных и обоснованных в [2, 4]. Абстрактной экспликацией модели-универсума ИН в контексте представления и моделирования знаний является k-гиперпространство семиотико-

¹ Работа выполнена при финансовой поддержке РФФИ, грант 02-07-90239в, и Министерства образования и науки

¹ Работа выполнена при финансовой поддержке РФФИ, грант 02-07-90239в, и Министерства образования и наукт РФ, НТП 2.1.3.7233.

множество ("цветов") с элементами $p \in P$, областями допустимых значений $p - \lambda(p)$ и выделенными i-ми значениями (в случае их существования) $\lambda_i(p) \in \lambda(p)$, есть процесс приписывания различных цветов из P элементам $g_{\eta\tau} \sqcap (V_\tau, E_{\eta\tau})$, когда каждому элементу может быть поставлено в соответствие некоторое подмножество цветов из заданного множества P. Гипертопограф $g_{\eta\tau}$, хроматизированный заданным множеством цветов, есть *хроматический* $\eta\tau$ -граф (X- $\eta\tau$ -граф) $g_{\eta\tau}X$. Допустимое множество Х- $\eta\tau$ -графов обозначается через $G^k_{\eta\tau}X \sqcap \{g_{\eta\tau}X\}$. Актуальная хроматизация информационных объектов проведена с использованием эпикуровой характеризации МС посредством универсальных свойств OP, аппарата "математической информатики" по А.Чечкину и концепции "информационной динамики" Н.Гладких (объекты-признаки). В рамках теории решеток и частично-упорядоченных множеств синтезирован ряд шкал свойств и дистрибутивных решеток понятий, в том числе объединенная базовая шкала частных, включая атомарные, свойств, и шкала УЭС (универсальных элементарных свойств).

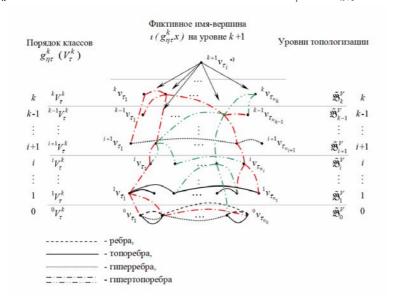
Семиотизация X- $\eta\tau$ -графов (по Д.Поспелову), осуществляется путем конструктивного поименования как элементов порождающих его множеств, так и непосредственно X- $\eta\tau$ -графа как элемента множества $G^k_{\eta\tau}x$. В результате каждый атрибут модели r получает собственное имя - $\iota(r)$, реализуемое в аксиоматике X- $\eta\tau$ -графов "цветом: именем" - $p\iota$. Числовой X- $\eta\tau$ -граф при этом есть частный случай семиотико-хроматического (СХ-) $\eta\tau$ -графа. Частными случаями СХ- $\eta\tau$ -графа являются СХ-графы, СХ-гиперграфы, СХ-топографы и СХ-паратопографы [5]. В свою очередь, частными случаями СХ-графов являются графы (помеченные, нумерованные, взвешенные, раскрашенные, вероятностные и т.п.), орграфы, мульти- и псевдографы, сети, S- и П-графы (семиотические) и т.д.

Семиотико-хроматическое k-гиперпространство CX- $\eta\tau$ -графов Γ S есть допустимое множество семиотико-хроматических $\eta\tau$ -графов $G^k_{\eta\tau} x$ уровня топологизации k, порождающие объекты представителей $g^k_{\eta\tau} x = (Vg^k_{\tau}, Eg^k_{\eta\tau}, Pg^k_{\tau}), \ g^k_{\eta\tau} x \in G^k_{\eta\tau} x$, $Vg^k_{\tau} \subseteq V^k_{\tau} \times Pg^{Vg^k_{\tau}}$, $Eg^k_{\eta\tau} \subseteq E^k_{\eta\tau} \times Pg^{Eg^k_{\eta\tau}}$, $Pg^k_{\tau} \sqcap Pg^{Vg^k_{\tau}} \cup Pg^{Eg^k_{\eta\tau}}$, которого, а именно V^k_{τ} и $E^k_{\eta\tau}$, есть элементы соответственно булеанов B^V_k и B^V_{k+1} k-го и k+1-го уровней топологизации множестваносителя V. Гипертопограф (монохромный) $g^k_{\eta\tau} \sqcap (V^k_{\tau}, E^k_{\eta\tau}), \ g^k_{\eta\tau} \in G^k_{\eta\tau}$ есть опорный $\eta\tau$ -граф k-го уровня топологизации CX- $\eta\tau$ -графа $g^k_{\eta\tau} x$ (см. рис. 1).

Синтез модели монохромного k-гиперпространства Γ S^m проведен с использованием категорий "тождества / различия" по Γ .Лейбницу и теоретико-множественной парадигмы Γ .Кантора путем последовательной топологизации множества-носителя гипертопографа V в булеан B^V_k уровня топологизации k ($V \sqcup \mathsf{B}^V_0 \subset \mathsf{B}^V_1 \subset \mathsf{B}^V_2 \subset ... \subset \mathsf{B}^V_k$). Булеан B^V_k уровня топологизации k, есть результат последовательной k-топологизации множества-носителя $\eta\tau$ -графа $V \sqcup \mathsf{B}^V_0$, V = n, когда на очередном этапе топологизации i+1 в качестве исходных неделимых и различимых элементов множества, порождающих булеан B^V_{i+1} , выступают непустые элементы булеана B^V_i . При включении в $\mathsf{B}^V_k \varnothing$ -множества $\mathsf{B}^V_k = 2^{2^{i-1}-1}$ (k экземпляров 2). Модель k-гиперпространства Γ S учитывает как особенности структуризации информационных объектов произвольного уровня вложенности, так

произвольные свойства и отношения элементов исследуемых структур. В общей постановке произвольная вершина ${}^iv_{\tau_j}$ есть "вложенный" в $g^k_{\eta\tau}$ $\eta\tau$ -граф ${}^ig_{\eta\tau}$ Термин "гиперпространство" задействован вследствие доказанного изоморфизма универсальных алгебр $\mathsf{A}_{\mathsf{B}^V_{k+1}} \ \cup \ <\mathsf{B}^V_{k+1}, \ (\cup, \cap) > \ \mathsf{u} \ \mathsf{A}_{[GF(2)]/\mathsf{B}^V_k/} = < [GF(2)]/\mathsf{B}^V_k/, \ (\bigvee, \bigwedge) > ,$ определенных относительно отображения $\varphi \colon X_i \ \longrightarrow \ X_i$, где X_i есть произвольный элемент булеана

 B_{k+1}^V (подмножество B_k^V) и \vec{X}_i есть элемент булева k-гиперпространства $[\mathit{GF}(2)]^{|\mathsf{B}_k^V|}$ размерности $|\mathsf{B}_k^V|$, при биективном соответствии монохромного $G_{\eta\tau}^k \longleftrightarrow \mathsf{B}_{k+1}^V \longleftrightarrow [\mathit{GF}(2)]^{|\mathsf{B}_k^V|}$.



Р и с. 1. Графическая иллюстрация синтеза опорного $\eta \tau$ -графа в k-гиперпространстве Γ s^m в иерархии порядка $g_{\eta \tau}^2$

Модель k-гиперпространства вложенных СХ-*ηт-графов* ГS поглощает вышеперечисленные модели представления знаний и посредством биективных алгоритмических процедур прямой и обратной редукции связана с семиотико-лингвистическими моделями последовательной коммуникации (по Дж. фон Неймана и К.Шеннону). Отношения порядка, определенные на Γ s^m и, параметричность интервала топологизации позволяет синтезировать семейство моделей вложенных СХ- $\eta\tau$ -графов, число уровней иерархии которых определяется требованиями детализации и четкости в условиях ограничений

Последующая алгебраизация модели (синтез сигнатуры) в форме одноосновной метаалгебры $\mathsf{A}_{G^k_{\eta \tau^X}} = A \, (\, G^k_{\eta \tau} x, \, \varPsi_{G^k_{\eta \tau^X}}) \,$ обеспечивает возможность моделирования динамических процессов

функционирования ИС. В свою очередь, введение ряда мер и метрик на элементах Γ S (в развитие концепций А. Колмогорова, Н. Рашевского, У. Эшби, Ф. Хаусдорфа) проецирует синтезированную модель на область измеримых метрических гиперпространств, что позволяет свести задачи количественного анализа прагматических атрибутов информации, воспринимаемой и преобразуемой ИС, к исследованию свойств k-гиперпространства Γ S.

Список литературы

("вычислимость") на их реализацию.

- 1. Баранович А. Е. Универсальный подход к структурному моделированию директивно-целевых информационных процессов // Сб. ст. - М., МО РФ, 1997.
- 2. Баранович А. Е. Семиотико-хроматические гипертопографы. Введение в аксиоматическую теорию: информационный аспект. - М., МО РФ, 2003.
- 3. Баранович А. Е. Исчисление ценности фактуальной информации в интеллектуальных системах (методологические вопросы) // Мат. межд. научн.-техн. конф. "Сист. пробл. качест., матем. модел., информ., электрон. и лазер. техн.". Ч. 4. - М.-Сочи, МИЭМ (ГТУ), 2004.
- 4. Baranovich A. E., Sidorov O. V. Basic principles of knowledge's representation and speech information processing within integrated intelligent system // Proc. of Intern. conf. "Speech & Computer" (SPECOM'2006). - St.-Petersburg, 2006.

² Топоребра в узком смысле есть ребра на топомножествах, в широком смысле - совокупности ребер на Топоребра в узком смысле есть ребра на топомножествах, в широком смысле - совокупности ребер на топомножествах фиксированного уровня топологизации. Ребра как частный случай топоребер и гиперребер присущи только $\tilde{\mathsf{B}}_0^V$, в то время как топоребра - любым $\tilde{\mathsf{B}}_i^V$,

 $i=\overline{0,k}$. Гиперребра характеризуют произвольные подмножества элементов топомножеств ${}^iV_{\tau}^k$ фиксированного уровня топологизации i, гипертопоребра - произвольные подмножества элементов B^{V}_{k} , входящих в $g^{k}_{\eta\tau}$ (любого уровня топологизации $i = \overline{0.k}$).

О параметризованном построении подобия ресурсов в сетях Петри 1

Башкин В. А.,

ст.преп., ЯрГУ им. П. Г. Демидова, Ярославль, Россия, E-mail: bas@uniyar.ac.ru

Ломазова И. А.

с.н.с., ИПС РАН, Переславль-Залесский, Россия, E-mail: irina@lomazova.pereslavl.ru

Ресурс определяется как часть разметки сети Петри. Два ресурса подобны, если замена одного из них на другой в любой разметке данной сети Петри не изменяет ее наблюдаемое поведение [1,3]. Подобие ресурсов неразрешимо, но при этом обладает конечным базисом.

В данной работе рассматриваются способы приближенного вычисления подобия ресурсов. Определяются три класса параметризованных отношений, тем или иным способом приближающих отношение подобия ресурсов. Первый класс — ограниченные подобия ресурсов. Рассматриваются только подобные пары, которые можно получить на основе базиса, состоящего из ресурсов ограниченной мощности. Второй класс — ограниченные бисимуляции ресурсов. Добавляется дополнительное требование — множество пар должно быть замкнуто относительно срабатывания переходов сети. Третий класс — расслоенные подобия ресурсов. В отличие от первых двух классов, это не сужения, а расширения подобия ресурсов. Ограничение накладывается не на ресурсы, а на поведение сети — учитываются только срабатывания последовательностей переходов ограниченной длины.

Исследуются свойства этих отношений. В частности, доказано, что все они обладают конечными базисами. При этом ограниченные подобия ресурсов неразрешимы, а ограниченные бисимуляции и расслоенные подобия ресурсов вычислимы и поэтому могут использоваться на практике в качестве параметризованных аппроксимаций подобия ресурсов.

Через Nat обозначим множество неотрицательных целых чисел. Множество всех мультимножеств над множеством X обозначим как $\mathcal{M}(X)$.

Пусть $(M_1, M_2), (M'_1, M'_2) \in \mathcal{M}(X) \times \mathcal{M}(X)$ – пары мультимножеств над X. Определим их сумму как

$$(M_1, M_2) + (M'_1, M'_2) =_{def} (M_1 + M'_1, M_2 + M'_2).$$

Для данного отношения $B \subseteq \mathcal{M}(X) \times \mathcal{M}(X)$ его аддитивным транзитивным замыканием B^{AT} назовем наименьшее (по вложению) подмножество множества $\mathcal{M}(X) \times \mathcal{M}(X)$, такое, что

- $1)B \subseteq B^{AT};$
- $\begin{array}{ll} -1 & -1 \\ 2) \forall (M_1,M_2), \ (M_1',M_2') \in B^{AT} & (M_1+M_1',M_2+M_2') \in B^{AT}; \\ 3) \forall (M_1,M_2), \ (M_2,M_3) \in B^{AT} & (M_1,M_3) \in B^{AT}. \end{array}$

Отношение $B' \subseteq \mathcal{M}(X) \times \mathcal{M}(X)$ называется AT-базисом отношения B, если $(B')^{AT} = B^{AT}$.

Сетью Петри называется набор N = (P, T, F), где P — конечное множество позиций; T конечное множество переходов, $P \cap T = \emptyset$; $F : (P \times T) \cup (T \times P) \to Nat$ — функция инцидентности.

Pазметкой (состоянием) сети N называется функция вида $M:P \to Nat$, сопоставляющая каждой позиции сети некоторое натуральное число (или ноль). Разметка — мультимножество над множеством позиций сети.

Для перехода $t \in T$ через • t и t • обозначим мультимножества его входных и выходных позиций, такие, что $\forall p \in P$ $\bullet t(p) =_{def} F(p,t), \ t^{\bullet}(p) =_{def} F(t,p).$ Переход $t \in T$ готов κ срабатыванию при разметке M, если $^{\bullet}t \subseteq M$. Готовый к срабатыванию переход t может cpa6omamb, порождая новую разметку $M' =_{def} M - {}^{\bullet}t + t^{\bullet}$ (используется обозначение $M \stackrel{t}{\to} M'$).

Помеченной сетью Петри называется набор N = (P, T, F, l), где (P, T, F) — сеть Петри, $l: T \to I$ Act — помечающая функция, Act — конечное множество имен (меток) действий.

Отношение $R\subseteq \mathcal{M}(P)\times \mathcal{M}(P)$ называется бисимуляцией разметок, если для любой пары $(M_1,M_2)\in R$ и для любого $a\in Act$:

• если $M_1 \stackrel{a}{\to} M_1'$, то $M_2 \stackrel{a}{\to} M_2'$, причем $(M_1', M_2') \in R$; и

¹Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант №06-01-00106) и Президиума РАН, программа "Фундаментальные проблемы информатики и компьютерных технологий" (проект 223).

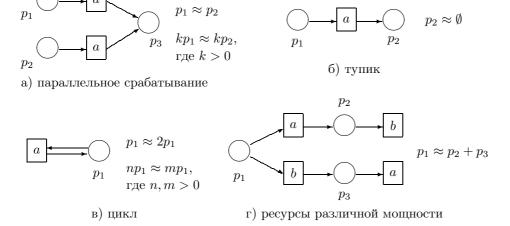


Рис. 1: Примеры подобных ресурсов

• если $M_2 \stackrel{a}{\to} M_2'$, то $M_1 \stackrel{a}{\to} M_1'$, причем $(M_1', M_2') \in R$.

Объединение всех бисимуляций разметок обозначается как \sim . Известно, что для любой сети отношение \sim является бисимуляцией разметок. Также известно [5], что бисимуляция разметок нераз-

Существует понятие расслоенной (stratified) бисимуляции разметок [6] (обозначается \sim_n). Оно определяется индуктивно:

Во-первых, полагаем $M_1 \sim_0 M_2$ для любых $M_1, M_2 \in \mathcal{M}(P)$. Далее, для любого $n \in Nat$ полагаем $M_1 \sim_{n+1} M_2$, если для любого $a \in Act$:

- если $M_1 \stackrel{a}{\to} M_1'$, то $M_2 \stackrel{a}{\to} M_2'$, причем $M_1' \sim_n M_2'$; и если $M_2 \stackrel{a}{\to} M_2'$, то $M_1 \stackrel{a}{\to} M_1'$, причем $M_1' \sim_n M_2'$.

Для любого n отношение \sim_n является отношением эквивалентности, кроме того, $(\sim_{n+1}) \subseteq (\sim_n)$. Бисимуляция разметок является пределом последовательности расслоенных бисимуляций: (\sim) = (\sim_{∞}) . Проблема n-расслоенной бисимулярности разметок разрешима для любого n.

Определение 1 Ресурсом обыкновенной сети Петри N = (P, T, F) называется мультимножество над множеством позиций Р.

Ресурсы являются частями разметок (в некотором смысле неделимыми), обеспечивающими то или иное поведение сети при любом еч состоянии.

Определение 2 (1,3) Пусть N = (P, T, F, l) – помеченная сеть Петри. Ресурсы r и s называются подобными (обозначается $r \approx s$), если для любой разметки $M \in \mathcal{M}(P)$ выполняется $M + r \sim M + s$.

Если ресурсы подобны, то в любой разметке мы можем заменить один из них на другой, и при этом дальнейшее наблюдаемое поведение системы не изменится (в смысле бисимулярности). Таким образом, подобие ресурсов отслеживает все ресурсы в сети Петри, обладающие идентичными свойствами при любом состоянии системы в целом.

Теорема 1 [1,3] Отношение подобия ресурсов в сетях Π етри

- 1) рефлексивно, симметрично, транзитивно и замкнуто относительно сложения пар мультимножеств;
- 2) обладает конечным АТ-базисом;
- 3) неразрешимо.

Пусть $n \in Nat$ — целое неотрицательное число, N = (P, T, F, l) — помеченная сеть Петри, (\approx) множество пар подобных ресурсов сети N.

Определение 3 Через ($\approx_{\overline{\eta}}$) обозначим подмножество множества (\approx), состоящее только из пар, получаемых аддитивным транзитивным замыканием элементов множества (\approx) , мощности левой и правой частей которых не превышают п:

$$(\approx_{\overline{n}}) =_{def} \{ (r,s) \in (\approx) \mid (|r| \le n \& |s| \le n) \lor (r,s) \in (\approx_{\overline{n}})^{AT} \}.$$

 $Om howehue \approx_{\overline{n}} hasoeem n$ -ограниченным подобием ресурсов.

Определение 4 Через $(\simeq_{\overline{n}})$ обозначим подмножество множества $(\approx_{\overline{n}})$, замкнутое относительно срабатывания переходов в сети:

```
(\simeq_{\overline{n}}) =_{def} \{ (r,s) \in (\approx_{\overline{n}}) \mid
         \forall t \in T : r \xrightarrow{t} r' \exists u \in T : s \xrightarrow{u} s' \& l(u) = l(t) \& (r', s') \in (\simeq_{\overline{n}});
         \forall u \in T : s \xrightarrow{u} s' \exists t \in T : r \xrightarrow{t} r' \& l(t) = l(u) \& (r', s') \in (\simeq_{\overline{n}}) \}.
```

 $Om howehue \simeq_{\overline{n}} назовем n$ -ограниченной бисимуляцией ресурсов.

Определение 5 Пусть $(\sim_n)-n$ -расслоенная бисимуляция разметок сети N. Через (\approx_n) обозначим множество пар ресурсов, задающее на (\sim_n) отношение подобия:

$$(\approx_n) =_{def} \{ (r,s) \in \mathcal{M}(P) \times \mathcal{M}(P) \mid \forall M \in \mathcal{M}(P) (M+r,M+s) \in (\sim_n) \}.$$

 $Om howehue \approx_n$ назовем n-расслоенным подобием ресурсов.

Теорема 2 Пусть N = (P, T, F, l) — помеченная сеть Петри. Тогда

- 1) $\forall n$ отношения $\approx_{\overline{n}}, \simeq_{\overline{n}} u \approx_n$ рефлексивны, симметричны, транзитивны и замкнуты относительно сложения пар мультимножеств;
- 2) $\forall n$ множества $(\approx_{\overline{n}}), (\simeq_{\overline{n}})$ $u (\approx_n)$ обладают конечными AT-базисами;
- 3) $\forall n$ выполняется $(\simeq_{\overline{n}}) \subseteq (\approx_{\overline{n}}) \subseteq (\approx) \subseteq (\approx_n);$
- 4) $\forall n$ выполняется $(\simeq_{\overline{n}}) \subseteq (\simeq_{\overline{n+1}}), \ (\approx_{\overline{n}}) \subseteq (\approx_{\overline{n+1}}), \ (\approx_n) \supseteq (\approx_{n+1});$ 5) $\exists n$ такое, что $\forall m \geq n$ выполняется $(\simeq_{\overline{m}}) = (\simeq), \ (\approx_{\overline{m}}) = (\approx);$
- 6) последовательность $\{(\approx_i)\}$ стремится κ (\approx) при $i\to\infty$, но не обязательно существует n, такое что $(\approx_n) = (\approx)$;
- 7) $\forall n > 0$ отношение $\approx_{\overline{n}}$ неразрешимо;
- 8) $\forall n$ отношения $\simeq_{\overline{n}} u \approx_n разрешимы;$
- 9) $\forall n$ множества $(\simeq_{\overline{n}})$ u (\approx_n) вычислимы.

Таким образом, последовательности $\{(\approx_i)\}$ и $\{(\approx_i)\}$ при $i\to\infty$ сходятся к одному и тому же пределу — подобию ресурсов (снизу и сверху соответственно). При этом (\approx_n) можно построить алгоритмически, а $(\approx_{\overline{n}})$ — нельзя. Однако можно построить важное сужение отношения $(\approx_{\overline{n}})$ ограниченную бисиммуляцию ресурсов ($\simeq_{\overline{n}}$). Доказано, что последовательность отношений $\{(\simeq_{\overline{i}})\}$ сходится к отношению, которое мы обозначили как (\simeq). Существует гипотеза, что (\simeq) = (\approx).

Список литературы

- 1. Башкин В.А., Ломазова И.А. Бисимуляция ресурсов в сетях Петри // Известия РАН: Теория и системы управления. 2003. №4. С.115–123.
- 2. Башкин В.А., Ломазова И.А. Подобие обобщенных ресурсов в сетях Петри // Труды МСО-2005. М.: МГУ. 2005. С.330-336.
- 3. Bashkin V.A., Lomazova I.A. Petri Nets and resource bisimulation // Fundamenta Informaticae. 2003. V.55. №2. P.101–114.
- 4. Bashkin V.A., Lomazova I.A. Similarity of Generalized Resources in Petri Nets // Proc. of PaCT'2005. Lecture Notes in Computer Science. 2005. V.3606. P.27-41.
- 5. P. Jančar. Decidability questions for bisimilarity of Petri nets and some related problems // Proc. of STACS'94. Lecture Notes in Computer Science. 1993. V.775. P.581–592.
- 6. Jančar P., Moller F. Techniques for decidability and undecidability of bisimilarity // Proc. of CONCUR'99. Lecture Notes in Computer Science. 1999. V.1664. P.30-45.

Рекуррентные нейронные сети в задачах распознавания образов

Бендерская Е. Н, доцент; Никитин К. В.,

Санкт-Петербургский государственный политехнический университет, 195251, Санкт-Петербург, ул. Политехническая, д. 29 bender@sp.ru, execiter@mail.ru

Выполнен обзор и проведена классификация основных моделей рекуррентных нейронных сетей, используемых для решения задач распознавания образов. Для исследования выбрана модель машины неустойчивых состояний как наиболее перспективная. Произведен анализ системы распознавания, построенной на основе машины неустойчивых состояний. Рассмотрены составляющие модели и их особенности. Приведены результаты имитационного моделирования работы машины неустойчивых состояний, решающей задачу распознавания случайных образов, представленных в виде пуассоновских потоков.

Ключевые слова: рекуррентные нейронные сети, распознавание образов, машина неустойчивых состояний, импульсные нейронные сети

Введение

Распознавание образов является одной из нетривиальных задач, выполняемых человеческим мозгом. Актуальность данной задачи высока в свете ее применения в различных управляющих системах и комплексах, получающих из внешней среды информацию и анализирующих ее. Наиболее интересным случаем является представление информации в форме образов чувственного восприятия — зрительных, звуковых и др.

Для решения задачи распознавания активно используется аппарат искусственных нейронных сетей (НС). Модели НС изначально были достаточно простыми по причине низкой производительности вычислительных устройств, что значительно ограничивало их возможности. С развитием вычислительных средств стали исследоваться все более сложные модели нейронных сетей. Модификация моделей НС развивается по нескольким направлениям: модификация структуры НС, нейронов, алгоритмов обучения. Наиболее значимой с точки зрения поведения и функционирования является модификация структуры нейронной сети. Исходя из структуры, все НС делятся на статические и динамические или на сети прямого распространения и рекуррентные сети.

Статические НС, как следует из названия, не имеют собственной динамики и их поведение полностью предсказуемо и определяется текущим входным воздействием. Данные НС описываются статическими функциональными зависимостями типа вход-выход. За счет этого такие НС могут быть легко обучены с помощью алгоритмов, так или иначе основанных на оптимизации. Однако статические НС имеют ряд существенных недостатков таких, как невозможность работы с изменяющимся входным сигналом и отсутствие памяти. Динамические или рекуррентные нейронные сети (РНС) лишены перечисленных недостатков, присущих статическим НС. За счет наличия обратных связей поведение рекуррентных сетей качественно изменяется и для их описания уже требуются дифференциальные или разностные уравнения. Сложность аналитического описания и исследования РНС поначалу затрудняло их развитие и применение для задач распознавания.

Существует большое количество разрозненных моделей РНС, но вопросам их классификации и сопоставления посвящена незначительная часть литературы по НС. Для применения РНС в задачах распознавания необходимо иметь системное представление об основных существующих моделях, их достоинствах, недостатках и областях применения.

1. Реккурентные нейронные сети и задача распознавания образов

Распознавание образов является задачей, в которой получаемый образ должен быть отнесен к одному из предопределенных классов. В тоже время задача распознавания образов связана с ассоциативной памятью, представляющей собой память, которая обучается на основе ассоциаций. Распознавание может быть рассмотрено, как ассоциативное соотнесение входного образа с известными или хранимыми.

Все новые подходы к решению данной задачи направлены на то, чтобы как можно подробнее и реалистичнее воссоздать процесс распознавания, который происходит в человеческом мозге. Исследования в этом направлении привели к использованию аппарата нейронных сетей в качестве базового для решения задач распознавания.

- РНС нейронные сети с одной или несколькими обратными связями. Поведение данных сетей описывается системой дифференциальных или разностных уравнений, а для обучения используются алгоритмы, учитывающие динамику функционирования НС. Практически все модели РНС могут быть разделены на два класса:
 - РНС с управляемой динамикой (в большинстве случае устойчивой). Для данных моделей РНС известны определенные зависимости, связывающие параметры НС со свойствами ее динамики. Задавая определенным образом параметры НС, можно добиваться нужных преобразований входных сигналов в выходные. Как правило, данные РНС работают с постоянными входными сигналамиобразами, задаваемыми как начальные условия.
 - РНС с неуправляемой динамикой. Данные модели РНС имеют достаточно сложную структуру, которая практически не поддается аналитическому описанию. Структура и параметры НС в одних вариантах подбираются случайным образом так, чтобы производительность была максимальной (LSM [21], ESM [14]). В других вариантах структура НС является фиксированной, а параметры находятся в результате обучения (LSTM [16]).

Рассмотрим каждый из двух вышеперечисленных классов РНС.

1.1. РНС с управляемой динамикой

РНС с управляемой динамикой синтезируются таким образом, что в установившемся режиме они могут находиться в определенном подмножестве отдельных своих состояний – аттракторов. Каждому из хранимых образов в этих НС соответствует определенный аттрактор и при задании в качестве начальных условий сети входного воздействия происходит переход НС в один из аттракторов, соответствующих запомненному образу, наиболее похожему на входной. Преимуществом данного подхода является то, что сеть практически всегда сходится к какому-либо аттрактору, поэтому можно говорить о завершенности процесса распознавания. Аттракторные РНС в силу своих свойств чаще используются для решения задач ассоциативной памяти. С точки зрения развития РНС с управляемой динамикой следует сказать, что основными учеными, развивающими это направление изначально были физики, исследовавшие коллективное поведение микросистем (Хопфилд [18], Хакен [5]).

Нейронная сеть Хопфилда

Одной из первых моделей аттракторных НС является сеть Хопфилда, имеющая физическую структуру, подобную спиновым стеклам. НС Хопфилда является полносвязной нейронной сетью с линейными симметричными обратными связями. Данная сеть обучается путем настройки весовых коэффициентов при помощи одного из вариантов правила Хэбба [15]. НС Хопфилда может быть использована для решения задачи автоассоциативной памяти: ассоциативной памяти, когда размерность входного и выходного образа совпадает.

Различают асинхронную и синхронную, дискретную и непрерывную модели сети Хопфилда. Для распознавания образов наибольшее распространение получил дискретный асинхронный вариант, при котором в каждый такт работы сети происходит расчет и изменение нового состояния только для одного из нейронов. В таком варианте НС Хопфилда является асимптотически устойчивой. Минимумам энергии НС, являющейся одновременно и функцией Ляпунова, соответствуют хранимые образы в случае их ограниченного количества.

Поскольку НС Хопфилда является одной из первых моделей РНС, ей присуще большое количество недостатков, среди основных можно назвать следующие: сравнительно небольшая информационная емкость; хранение ложных образов, «химер», являющихся наряду с истинными образами аттракторами НС; длительное время распознавания.

Было предложено большое количество модификаций сети Хопфилда, отличающихся от нее в какихлибо деталях. Все видоизменения НС Хопфилда можно разделить на следующие группы:

- 1. Модификация обучающего правила (обучение по генетическим алгоритму [13]).
- 2. Модификация элементов сети (использование многомерных [23], стохастических моделей нейронов [24, 25], изменение функции активации обычных нейронов).
- 3. Модификация алгоритма функционирования сети (изменение количества одновременно изменяемых за один такт нейронов, а также правила их выбора [11]).
- 4. Модификация структуры сети (использование асимметричных связей, ограничение на их число и тип [10]).
 - 5. Задание определенного кодирования информации (например, разреженного [3]).

Все вышеперечисленные способы так или иначе изменяют функцию энергии сети. Однако все их отличия от оригинальной НС Хопфилда являются локальными в том смысле, что они направлены на исправление некоторых ее недостатков, но общая концепция остается той же, и в целом каких-то принципиально новых свойств у этих моделей не появляется. Это замечание справедливо и для двунаправленной ассоциативной памяти (ВАМ, Bidirectional Associative Memory) [2]. В отличие от нейронной сети Хопфилда данная модель позволяет реализовать гетероассоциативную память, в которой

размерность входного и выходного образов отличаются. В остальном, основная идея и свойства ВАМ мало отличаются от модели Хопфилда.

Можно выделить два направления развития РНС с управляемой динамикой:

- «улучшение» управления динамикой, выражающееся в более четких алгоритмах обучения, возможности управления процессом распознавания;
- усложнение динамики, использование более сложных аттракторов, хаотической динамики, реализация идеи синхронизации как перехода от хаотического поведения к регулярному в ответ на определенный входной образ.

Синергетический компьютер Хакена

К представителям первого направления можно справедливо отнести модель, предложенную немецким ученым Г. Хакеном – так называемый синергетический компьютер [5]. В работах Хакена просматривается аналогия между процессом распознавания образов и образованием структур – параметров порядка, управляющих распознаванием. Динамика синергетического компьютера базируется на минимизации энергии, как и в сети Хопфилда. Особенностью является отсутствие ложных состояний и возможность описания процесса распознавания образов на макроуровне - через динамику параметров порядка. Отсутствие ложных образов реализуется за счет специально построенного энергетического функционала, что на практике осуществляется путем использования нелинейных обратных связей.

Синергетический компьютер, в отличие от множества моделей аттракторных НС, позволяет влиять на процесс распознавания с помощью параметров внимания. Манипулируя этими параметрами, можно концентрировать внимание распознающей системы на определенных образах по аналогии с тем, как это происходит у человека при распознавании сложных, неоднозначных образов (фотографий группы людей).

Управление распознаванием в синергетическом компьютере может быть достаточно просто использовано и применено только на макромодели, работающей с параметрами порядка. Для микромодели такой простоты не наблюдается. Поэтому в большинстве приложений используется именно модель синергетического компьютера на параметрах порядка.

Модели РНС со сложной динамикой

Существует ряд моделей РНС с простыми точечными аттракторами, но со специфичными нелинейными связями, дающими им большие возможности по хранению информации. Примером может служить релаксационная модель ассоциативной памяти [8], основанная на потенциале кулона. В целом использование аттракторов в форме фиксированных точек значительно сужает информационный объем, заключенный в динамике изменения состояния НС. Поэтому наблюдается определенная тенденция к синтезу динамических НС, имеющих более сложные аттракторы типа предельный цикл, тор или псевдохаотический аттрактор. Использование сложных аттракторов значительно увеличивает возможности НС по распознаванию зашумленных изображений.

Отдельным вариантом моделей РНС со сложной динамикой является модель, предложенная в [1]. В данной модели осуществляется запись информации на одномерные и многомерные отображения в форме предельных циклов. Каждый из предельных циклов представляет собой запоминаемое слово. Благодаря данному подходу возможно ассоциативное восстановление информации по неполным данным. Развитием подхода является реализация идеи образования порядка из хаоса [7]. В исходном состоянии отображение неустойчиво и генерирует хаотическую последовательность. При подаче входного сигнала отображение становится устойчивым, а сама система выходит на предельный цикл, соответствующий запомненному образу и наиболее близкому к входному.

Проблемным моментом использования сложной динамики в функционировании НС является извлечение нужной информации. К примеру, при использовании точечных аттракторов по конечному состоянию можно определить распознанный образ. При использовании аттракторов типа предельный цикл необходимо производить анализ траектории изменения состояния сети. В случае же сложных регулярных и псевдохаотических аттракторов извлечь информацию из состояния становится еще сложнее. Поэтому перед тем, как начать использовать сложную динамику должны быть синтезированы алгоритмы для извлечения из нее полезной информации.

Осцилляторные модели нейронных сетей

Возможность моделирования работы нейронных сетей, состоящих из элементов, подобных биологическому прототипу нейрона и являющихся по своей структуре рекуррентными, породило новые распознающие и ассоциативные системы. В частности, одним из таких примеров являются осцилляторные нейронные сети [9, 17]. Данный класс нейронных сетей предполагает импульсновременное кодирование информации нейронами. Процесс распознавания определяется как синхронные колебания определенных групп нейронов. Было предложено ряд моделей [20], базирующихся на синаптической пластичности Хэбба, в чем-то похожих на сети Хопфилда, но кодирующих информацию в

пространственно-временной форме и являющихся более схожими с биологическими прототипами. По свойствам и возможностям данные модели демонстрируют при распознавании лучшие свойства.

1.2. РНС с неуправляемой динамикой («неаттракторные» НС)

Как уже говорилось выше, при использовании аттракторных PHC со сложной динамикой возникают определенные затруднения, связанные с извлечением из них информации. Трудность заключается в определении, какому из аттракторов принадлежит текущая последовательность состояний HC. Идея PHC с неуправляемой динамикой заключается в отказе от «аттракторного мышления», предполагающему наличия соответствия каждого образа своему аттрактору. В результате PHC рассматривается как динамический фильтр, генерирующий определенный выходной сигнал в ответ на изменяющийся входной. При этом за счет обратной связи и нелинейности входная динамика определенным образом фиксируется во внутреннем состоянии сети.

Важно подчеркнуть принципиальное отличие рассматриваемого подхода от подхода, используемого в аттракторных НС. Аттракторные НС изначально ориентированы на работу со статическим входным сигналом. Данный же вариант сетей в качестве входных образов получает изменяющийся во времени сигнал. Причем сходимость сети к какому-либо аттрактору не требуется — важно, чтобы сеть разделяла различные входные последовательности, а в случае их зашумленности эффективно производила аппроксимацию.

Все РНС с неуправляемой динамикой могут быть либо обучаемыми, либо необучаемыми. Рассмотрим отдельно каждую из групп.

Обучаемые РНС

Обучаемые РНС могут быть непосредственно обучены на воспроизведение необходимого отображения путем модификации межнейронных связей. Для их обучения в настоящее время существует два основных алгоритма [4]:

- обратное распространение во времени (Back Propagation Through Time, BPTT);
- рекуррентное обучение в реальном времени (Real Time Recurrent Learning, RTRL).

Эти алгоритма имеют ряд ограничений по применению и требуют больших вычислительных затрат [16]. Как один из способов решения проблемы, в [16] предлагается использовать специальную модель – LSTM (Long Short-Term Memory) – так называемую обобщающую кратковременную память. Она позволяет успешно применять выше описанные алгоритмы за счет специфической структуры составляющих ее элементов. Модель LSTM может быть применена для распознавания непрерывных временных сигналов (например, речи). Ее недостатком является большая чувствительность к шуму во входных данных. В частности, уже при небольшом уровне шума, модель теряет большинство своих полезных свойств.

В общем можно сказать, что существует большое количество ограничений по использованию обучаемых РНС с неуправляемой динамикой в задачах распознавания. И эта ограниченность связана прежде всего с противоречивыми требованиями к НС: с одной стороны по принципу построения это «неаттракторные» НС (НС с неуправляемой динамикой), а с другой стороны их необходимо обучать формировать определенную выходную последовательность в ответ на входную, т.е. часть из возможных последовательностей состояний должна быть управляемой.

Необучаемые РНС

Необучаемые РНС с неконтролируемой динамикой используются совместно со считывающими модулями — наблюдателями, выполняющими функцию идентификации сигнала с РНС. При этом структура и параметры РНС являются фиксированными и не изменяются в процессе обучения. РНС используется для генерации определенного уникального состояния в ответ на входную временную последовательность. Это состояние является обучающим сигналом считывающих модулей, которые настраиваются на извлечение определенной информации из него. Необучаемые РНС с неконтролируемой динамикой составляют основу двух похожих моделей, предназначенных для распознавания входных временных сигналов.

Первая модель называется ESM (Echo State Machine) — машина эхо-состояний. Она состоит из обычных нейронов и функционирует в дискретном времени. При подаче на входы ESM определенной временной последовательности она переходит в так называемое «эхо-состояние». Причем для разных входных последовательностей это «эхо-состояние» уникально. Выходные сигналы с ESM подаются на входы считывающих модулей, которые могут быть обучены на выделение определенных «эхо-состояний» и, таким образом, на распознавание конкретных образов. Наличие у ESM таких свойств, как разделение входных последовательностей и аппроксимация, позволяет успешно распознавать зашумленные входные образы.

Вторая модель по логике работы аналогична модели ESM, но в качестве составляющих элементов в ней используются импульсные нейроны. Эта модель называется LSM (Liquid State Machine) – машина неустойчивых состояний. За счет использования импульсных нейронов данная модель является более

биологически-ориентированной, в отличие от ESM. По характеру своего поведения LSM напоминает жидкость в том смысле, что все входные возмущения отражаются в динамике ее внутреннего состояния по аналогии с жидкостью.

По структуре LSM представляет из себя рекуррентную сеть, состоящую из импульсных нейронов. Данная сеть позволяет обрабатывать непрерывно изменяющуюся информацию на ее входах. При этом за счет рекуррентности и нелинейности входная динамика определенным образом фиксируется во внутреннем состоянии сети. Задача считывающих модулей заключается в обучении извлечения фиксированной входной динамики из последовательности внутренних состояний сети. Другими словами, задачей считывающих модулей является идентификация по возмущениям динамической системы (которые в общем случае неустойчивы) устойчивого состояния, или образа, который нужно распознать.

Структура и параметры LSM не являются фиксированными и могут быть выбраны исходя из обеспечения лучших критериев качества распознавания. В работах [19] данная модель рассматривается, как универсальное вычислительное устройство на входных импульсных потоках по аналогии с машиной Тьюринга, произвольно соединенными булевыми сетями и клеточными автоматами.

2. Машина неустойчивых состояний

Модель LSM структурно является РНС, элементами которой являются импульсные нейроны. Другими словами, LSM является импульсной нейронной сетью. В силу относительной новизны импульсных нейронных сетей, а также их важности для описания модели LSM, вначале рассмотрим модели импульсных нейронов и импульсных сетей. После этого будет детально описана модель LSM и система распознавания на ее основе.

2.1. Импульсные нейронные сети

Импульсные НС являются биологически ориентированными НС. В отличие от традиционных НС они передают информацию не в форме усредненного значения нейронной активности, а через последовательности импульсных сигналов. Выходной сигнал нейрона состоит из коротких электрических импульсов (также называемых действующими потенциалами или «спайками»), имеющих амплитуду 100 мВ и длительность 1-2 мс. Форма импульсов не изменяется при передаче по аксону. Цепь действующих потенциалов, вызванных одним нейроном, называется импульсной последовательностью — чередой одинаковых событий, возникающих в определенные или случайные моменты времени.

Поскольку все генерируемые импульсы имеют примерно одинаковую форму, то информация, скорее всего, содержится не в форме импульсов, а в их числе и в точных временах их возникновения. Минимальное временное расстояние между двумя импульсами называется абсолютным рефракционным периодом нейрона.

Модели импульсных нейронов

Существует множество различных моделей [12] импульсных нейронов. Наиболее детализированной и сложной является модель Ходкина-Хаксли. Она основана на экспериментальном исследовании большого количества нейронов кальмара. Система дифференциальных уравнений данной модели описывает точную реакцию потенциала мембраны в ответ на различные входные воздействия. Однако, такая реалистичность приводит к большим вычислительным затратам, в результате чего модель не очень подходит для экспериментов с НС, состоящими из большого количества нейронов.

Поскольку целью работы является исследование распознающей системы на основе LSM, то в качестве модели импульсного нейрона используется более простая модель под названием «обобщение-отклик» (Integrate-and-Fire, IaF). Данная модель является частным случаем обобщенной модели импульсного отклика (Spike Response Model, SRM). Момент f генерации импульса у i нейрона обозначается через $t_i^{(f)}$. Импульсная последовательность, генерируемая нейроном определяется следующим образом: $F_i = \{t_i^{(1)}, \dots, t_i^{(n)}\}$. Переменной u_i обозначается потенциал мембраны или внешнее состояние i нейрона. Если потенциал мембраны нейрона достигает порогового значения g_i , то генерируется импульс. Формально условие для генерации импульсов i нейроном выглядит следующим образом: $f_i = \{t \mid u_i(t) = g \& u_i(t) > 0\}$.

Модель «обобщение-отклик» импульсных нейронов

В модели IaF импульсы рассматриваются как короткие импульсные токи. Если говорить точнее, то каждый импульс моделируется δ -функцией Дирака. После того, как импульс приходит на синапс, мгновенно заряжаются все связанные с ним постсинаптические нейроны. Это изменение напряжения называется постсинаптический потенциал. По достижению потенциала мембраны порогового значения, он сбрасывается и генерируется новый импульс.

Динамика изменения потенциалов мембраны IaF нейрона до формирования импульса описывается следующим дифференциальным уравнением:

$$\tau_{m} \frac{\partial u}{\partial t} = -(u(t) - U_{rest}) + R \cdot I(t), \qquad (1)$$

где $^{\tau_m}$ - постоянная времени, R –входное сопротивление, I(t) – суммарный входной ток, $^{U_{rest}}$ – потенциал покоя нейрона. Для учета периода рефракции в течение этого периода значение потенциала остается постоянным и равным 0 вне зависимости от входного тока. После возбуждения нейрона его

потенциал становится равным так называемому потенциалу сброса $^{U_{\it reset}}$.

Суммарный входной ток IaF нейрона определяется следующим выражением:
$$I_{i}(t) = i^{ext} + \sum_{j \in \bar{A}i} c_{ij} \sum_{t_{j}^{(f)} \in F_{j}} \delta\left(t - t_{j}^{(f)} - \Delta^{ij}\right), \tag{2}$$

где $^{C_{ij}}$ - значение силы синаптической связи между і и ј нейроном, Δ^{ij} - синаптическая задержка при передаче импульса от і к і нейрону, а i^{ext} - значение внешнего шумового воздействия на нейрон.

Импульсный нейрон называется возбуждающим, если все исходящие от него синаптические связи являются положительными. В противном случае, когда все синаптические связи у нейрона отрицательные, он называется подавляющим. Как видно, ІаF модель импульсного нейрона несмотря на свою относительную простоту, намного сложнее частотной модели нейрона. Ее достоинством является достаточно простое аппаратное воплощение.

Линамические синапсы

Помимо преобразования сигналов в самих импульсных нейронах важный вклад в обработку информации вносят синапсы [19]. В действительности они действуют как динамические запоминающие устройства, изменяющие свои передающие свойства в зависимости от недавно переданной импульсной последовательности. Несмотря на то, что большинство моделей НС используют синапсы как статические взвешивающие функции, в ряде работ было показано, что синапсы, моделирующие более сложные динамические свойства, являются более мощным вычислительным средством.

Такие синапсы получили название динамические. Не вдаваясь в биологические подробности механизма работы динамических синапсов, остановимся на одной из моделей, формально иллюстрирующей данный механизм. Данная модель была предложена Маркрамом [22] и де-факто является наиболее повсеместно используемой. Она описывается следующими уравнениями:

$$w_n = A_n = A \cdot u_n \cdot R_n \tag{3}$$

$$u_{n+1} = U + u_n (1 - U) \exp(-\frac{\Delta t_n}{F}),$$
 (4)

$$R_{n+1} = 1 + (R_n - R_n u_n - 1) \exp(-\frac{\Delta t_n}{D}), \tag{5}$$

$$u_1 = U, R_1 = 1 (6)$$

Четыре основных параметра модели – A, U, F и D. Основной параметр A_n является силой синаптической связи, изменяющейся по приходу импульсов. Остальные параметры описывают специфику механизма динамических синапсов и их подробное описание дано в работе [22].

Использование импульсных НС

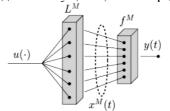
Импульсные нейронные сети имеют множество свойств, позволяющих рассматривать их отдельно от традиционных НС с частотным кодированием. За счет временного кодирования импульсные НС теоретически подходят для решения задач обработки временных входных сигналов. Нелинейная реакция на входной сигнал дает данным НС более высокие вычислительные возможности. Однако проблемным моментом является использование сложной динамики этих нейронных сетей. В частности, до сих пор не было предложено эффективного алгоритма декодирования многомерного импульсного сигнала. Также нерешенным является вопрос обучения данных нейронных сетей. Новым подходом к решению проблем импульсных НС является гипотеза о том, что источник вычислительной мощи этих НС лежит в использовании их динамики. Поэтому задача управления динамикой НС отходит на второй план.

2.2. Система распознавания на основе машины неустойчивых состояний

Независимо друг от друга Маасс [21] и Ягер [14] предложили схожие по общей идее модели: машину неустойчивых состояний (LSM) и машину эхо-состояний (ESM). Эти две похожие друг на друга математические модели являются универсальными вычислительными устройствами, работающими с непрерывным входным сигналом в реальном времени. В обеих моделях входной сигнал обрабатывается РНС. За счет сложности внутренней динамики РНС ее состояние может содержать важную информацию обо всех прошлых входных сигналах. Несмотря на то, что это состояние может быть непонятно человеку, его может успешно распознать и классифицировать считывающим устройством (наблюдателем), способным извлекать значимую информацию о прошлых входных сигналах из состояния РНС.

Математическая модель системы распознавания на основе машины неустойчивых состояний

Как и машина Тьюринга, модель машины неустойчивых состояний основана на строгой математической структуре, гарантирующей при идеализированных условиях универсальную вычислительную мощность. Модель LSM в отличие от машин Тьюринга ориентирована на вычисления в реальном времени. Входная функция u(t) может быть непрерывной последовательностью возмущений, а целевая выходная функция — некоторой функцией времени y(t), обеспечивающей учет изменения входной функции. Для того, чтобы машина (М) отобразила входные функции времени в выходные функции времени, предполагается, что она генерирует в каждый момент времени внутреннее состояние $x^M(t)$, являющееся ее откликом на входные возмущения, т.е. на предыдущие значения входного сигнал



u(s), $s \le t$ (см рис. 1 - Архитектура LSM.).

Входная функция времени $u(\cdot)$ подается на вход фильтра L^M и формирует на его выходе состояние $x^M(t)$, которое затем преобразуется считывающим отображением без памяти f^M для формирования выходного сигнала y(t) [21]. В математических терминах состояние М может быть определено через некоторый оператор или фильтр L^M , отображающий входные функции $u(\cdot)$ в функции $x^M(t)$, представляющие траектории внутреннего состояния:

$$x^{M}(t) = (L^{M}u)(t) \tag{7}$$

Второй компонентой M является считывающее отображение без памяти f^{M} , преобразующее в каждый момент времени t текущее состояние $x^{M}(t)$ в выходной сигнал:

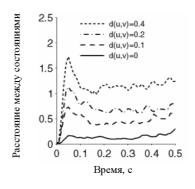
$$y(t) = f^{M}(x^{M}(t)). \tag{8}$$

Так как считывающее отображение не имеет памяти, вся информация о входах $\mathbf{u}(\mathbf{s})$ должна содержаться во внутреннем состоянии $x^{M}(t)$.

Основные свойства модели машины неустойчивых состояний

В основу системы распознавания заложены два ключевых свойства: разделение и аппроксимация. Свойство разделения определяется как различие между двумя траекториями движения системы при различных входных последовательностях. Как видно из определения, свойство разделения реализуется непосредственно в LSM. Свойство аппроксимации характеризует разрешающие и кодирующие возможности считывающих модулей. Оно определяется как способность отличать и преобразовывать различные внутренние состояния LSM в целевые выходные значения. В то время как разделяющее свойство больше зависит от сложности LSM, аппроксимирующее свойство зависит сильнее всего от адаптации считывающего механизма к конкретной задаче.

На рис. 2 приведена зависимость среднего расстояния между двумя состояниями LSM при различных расстояниях между входными последовательностями. С ростом расстояния между входными последовательностями увеличивается расстояние между двумя состояниями, что говорит о реализации моделью LSM свойства разделения.



Расстояние между двумя состояниями растет с увеличением расстояния d(u,v) между двумя импульсными последовательностями и v [21].

Для определения различия между двумя состояниями Maacc, автор LSM [21], предлагает использовать усредненную евклидову норму:

$$d_{s}(t,u,v) = \|X_{u}^{M}(t) - X_{v}^{M}(t)\|_{t},$$
(9)

где $X_u^M(t)$ и $X_v^M(t)$ - состояния LSM в момент времени t для входных сигналов u и v соответственно.

Реализация модели и ее параметры

LSM является импульсной PHC. Структурна этай PHC - трехмерноя решетка соединенных импульсных нейронов (рис. 3).

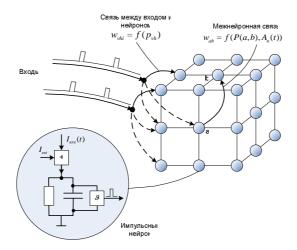


Рис. 3. Схематическое представление машины неустойчивых состояний в форме трехмерной решетки, состоящей из импульсных нейронов.

Общее число нейронов n_{nm} определяется размерностями решетки по осям X, Y и Z: n_X , n_Y , n_Z : $n_{nm} = n_X \cdot n_Y \cdot n_Z \ . \tag{10}$

Определенное число нейронов, составляющее p_{inh} от общего числа нейронов произвольно выбирается подавляющими. Остальная часть нейронов выбирается возбуждающими.

Связи между нейронам формируются на основе их взаимного расположения на решетке. При этом используется следующая формула, определяющая вероятность связей между двумя нейронами

$$P(a,b) = C(a,b) \cdot \exp\left(\left(-D(a,b)/\lambda\right)^{2}\right). \tag{11}$$

где буквами а и b обозначаются два нейрона, C(a,b) – постоянная, зависящая от типов взаимодействующих нейронов (возбуждающий или подавляющий), D(a,b) – евклидово расстояние между нейронами. Параметр λ одновременно управляет числом связей и средней длиной связи между нейронами.

Каждый вход LSM случайно соединен с определенным числом ее нейронов, составляющим p_{vh} от общего числа нейронов.

В качестве самих импульсных нейронов используются нейроны, описываемые IaF моделью. Их параметрами являются: порог срабатывания ${\cal G}$, сопротивление R , постоянная времени ${}^{\tau}$, период рефракции ${}^{t_{refr}}$, потенциал покоя ${}^{U_{rest}}$ и потенциал сброса ${}^{U_{reset}}$.

Для дополнительного учета входной динамики в состоянии LSM в качестве модели синапсов используется модель динамических синапсов. Помимо этого для внесения случайности, наблюдаемой в биологических нейронах, на вход каждого нейрона подается постоянный ток i^{ext} . Значения i^{ext} для каждого нейрона случайным образом берутся из нормального распределения.

Далее рассмотрим реализацию считывающих модулей. Каждый считывающий модуль представляет собой один или несколько элементов, на вход которых подается информация о состоянии LSM. В

качестве элементов, составляющих считывающий модуль, могут быть использованы импульсные, частотные или какие-то другие модели нейронов.

Особенностью использования неимпульсных нейронов является необходимость преобразования импульсной информации о состоянии LSM в непрерывную форму. Такое преобразование может быть выполнено искусственно путем фильтрации импульсного сигнала. Простейшим вариантом такой фильтрации является подсчет импульсов на выходах LSM за определенный промежуток времени – так

называемое временное окно t_{window} .

Для обучения неимпульсных нейронов может быть использован один из известных алгоритмов. Если используется пороговая модель нейрона, то подходит алгоритм на основе дельта-правила. В случае модели нейрона с линейной или сигмоидальной функцией активации можно воспользоваться одним из регрессионных алгоритмов. Для специфических моделей нейрона, таких как, например, параллельный персептрон, существуют свои индивидуальные эффективные обучающие правила и алгоритмы.

Представление информации на входе и выходе системы распознавания

Информация на входе системы распознавания представлена в форме импульсных пространственновременных образов. Пространственный характер образов определяется наличием нескольких входов, а временной — асинхронностью поступления импульсов на различные входы. Таким образом, LSM ориентирована на работу с изменяющимися временными образами. Примером таких образов может быть звуковой сигнал или изменяющееся во времени изображение.

Состояние LSM используется в качестве входного сигнала для считывающих модулей. При этом, как уже говорилось в предыдущем разделе, если считывающий модуль составлен из обычных нейронов, производится фильтрация состояния LSM с целью преобразования импульсного сигнала в непрерывный. В дальнейшем остановимся именно на этом варианте.

Выходной сигнал со считывающего модуля может быть дискретным или непрерывным в зависимости от используемой модели нейронов и решаемой задачи. В случае классификации целесообразно использовать дискретную модель нейрона. Если требуется выполнение какого-либо преобразования входного сигнала, то лучше подходит непрерывная модель. Важным является то, что все считывающие модули не зависят друг от друга, за счет чего может осуществляться параллельное решение разных задач.

Этапы функционирования системы распознавания на основе LSM

Функционирование системы распознавания можно разбить на 2 этапа: обучение и использование (или тестирование). На любом этапе функционирования на входы LSM последовательно подаются входные обучающие образы. Каждый образ подается в течение определенного времени так, чтобы в ответ на него успело сформироваться соответствующее уникальное состояние LSM. В конце подачи входного образа состояние LSM путем фильтрации преобразуется в вектор непрерывных величин и подается на вход считывающих модулей. Далее в зависимости от этапа функционирования производится либо изменение параметров считывающих модулей в соответствии с обучающим правилом либо только вычисление их выходных значений. Вся статистика по качеству распознавания образов собирается на этапе тестирования, когда параметры считывающих модулей остаются фиксированными.

Считывающие модули и алгоритмы их обучения

Считывающие модули могут быть построены из произвольного набора обучаемых элементов. При использовании в качестве элементов неимпульсных нейронов требуется производить фильтрацию выходного сигнала LSM. В данной работе базовыми считывающими модулями были выбраны и реализованы: пороговый персептрон [4] и параллельный персептрон [6].

3. Экспериментальное исследование

Задача классификации случайных потоков позволяет исследовать возможности системы по распознаванию нестационарных входных образов. Несмотря на свою случайность каждый из образов имеет хотя бы один отличный от других постоянный параметр, определяющий его свойства. В данном эксперименте в качестве входных образов будут использоваться случайные пуассоновские потоки. Параметром, отличающим один образ от другого будет закон изменения интенсивности потока во времени. Первый образ представлен пуассоновским потоком с частотой 20 Гц, второй образ также представлен пуассоновским потоком и его частота равна 10, 20, 50, 75, 100 и 200 Гц. Длительность представления входного образа равна 200 мс, длительность временного окна равна 50 мс. Результаты исследования приведены на рис. 4.

Считывающий нейрон обучается таким образом, чтобы выделять только один входной образ (выходное значение 1) и не выделять все остальные (выходное значение -1). Для обучения может быть использован как алгоритм с учителем (дельта-правило), так и алгоритм без учителя (кластеризация). Входные образы являются сами по себе случайными, поэтому нет необходимости добавлять в них дополнительную шумовую составляющую. Как видно из результатов, система успешно справляется с непростой на первый взгляд задачей разделения случайных образов. При одинаковых параметрах случайных образов (частота равна 20 Гц) система закономерно распознает каждый из образов с примерно

одинаковой вероятностью правильного распознавания 50 %. С увеличением различия параметров случайных образов вероятность правильного распознавания растет. Ошибка II рода превалирует над ошибкой I рода, что соответствует тому, что считывающий нейрон чаще выдает неправильное распознавание второго образа. Однако, правильное распознавание этот нейрон больше производил также для второго образа. Таким образом в среднем получалось, что на выходе два образа появлялись примерно одинаковое число раз. Кроме того, был проведен эксперимент по определению влияния длительности временного окна на качество распознавания. Экспериментальные результаты и теоретически максимальные значения вероятностей правильного распознавания случайных образов (рассчитаны с помощью специальной характеристики пуассоновских потоков – спектрального состава количества импульсов в потоке для заданного временного интервала) представлены на рис.5

Вероятность правильного распознавания случайных образов в исследуемой системе соотносится и даже превосходит теоретические значения максимальной вероятности распознавания пуассоновских потоков. Об этом говорят значения максимальной вероятности правильного распознавания, полученные в отдельных экспериментах. Среднее значение вероятности правильного распознавания у системы чуть ниже максимального теоретического значения, что вызвано большим количеством случайных факторов (от случайности в задании LSM до случайности образов).

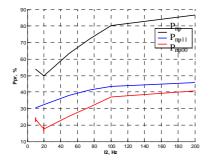


Рис. 4. Зависимость MO вероятностей правильного распознавания от частоты пуассоновского потока, соответствующего второму образу.

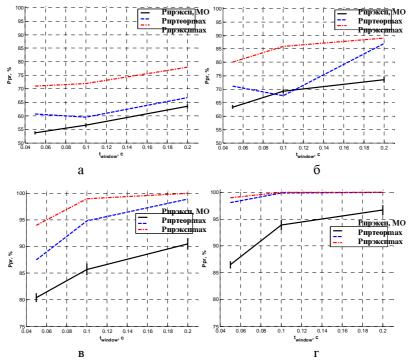


Рис. 5. Зависимость теоретических и экспериментальных показателей качества распознавания от длительности временного окна и частот пуассоновских потоков, соответствующих двум входным образам (первый образ представлен пуассоновским потоком с частотой 20 Γ ц, второй соответственно – для а)-10 Γ ц, б)- 50 Γ ц, в)- 100 Γ ц, г)-200 Γ ц.)

По полученнным экспериментальным данным можно утверждать, что система распознавания на основе машины неустойчивых состояний может успешно выполнять задачи, связанные с распознаванием изменяющихся во времени образов, представленных в импульсной форме. Причем поскольку LSM обладает кратковременной памятью, временные образы должны быть представлены на ограниченном

временном интервале, определяющемся глубиной памяти. Система также может использоваться для распознавания нестационарных, например, случайных образов, а также выполнять несколько задач параллельно. Преимуществом LSM по сравнению с остальными моделями является скорость распознавания в сочетании с возможностью работы в условиях зашумленных входных образов.

Список литературы

- 1. Андреев Ю. В., Дмитриев А. С., Куминов Д. А. Хаотические процессоры. // Успехи современной радиоэлектроники. -1997. N10. c. 50-79.
- 2. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.
- 3. Фролов А. А., Гусек Д., Муравьев И. П. Информационная эффективность ассоциативной памяти типа Хопфилда с разреженным кодированием. // Труды научных сессий МИФИ. Научная сессия МИФИ-2003. Ч.1 Нейроинформатика-2003. – с. 28-71.
- 4. Хайкин С. Нейронные сети: полный курс, 2-е издание. М.: Издательский дом «Вильямс», 2000. 1104 с.
- 5. Хакен Γ . Принципы работы головного мозга: Синергетический подход к активности мозга, поведению и когнитивной деятельности. М.: ПЕР СЭ, 2001. 351 с.
- 6. Auer P., Burgsteiner H. M., Maass W. The p-Delta Rule for Parallel Perceptrons. Submitted for publication. 2002.
- 7. Andreyev Y. V., Dmitriev A. S., Matveyev M. A. Application of chaotic dynamical systems to the problems of recognition and classification. // Proceeding of the 3-rd International Specialist Workshop on Nonlinear Dynamics of Electronic Systems (NDES-95). 1995. pp. 249-252.
- 8. Bachmann C. M., Cooper L. N., Dembo A. A relaxation model for memory with high storage density. // Proceedings of the National Academy of Sciences. 1987. vol. 84. pp. 7529-7531.
- 9. Brown A. G., Collins S. Innovative Idea: An Oscillatory Associative Memory. // Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems. 1994. pp. 186-191.
- 10. Davey N, Adams R. High capacity associative memories and connection constraints. // Connection Science. -2004. vol. 16. pp. 47-65.
- 11. Feng J., Brown D. Fixed-point attractor analysis for a class of Neurodynamics. // Neural Computation. 1998. vol. 10. pp. 189-213.
- 12. Gerstner W., Kistler W.M. Spiking Neuron Models. Single Neurons, Populations, Plasticity. Cambridge University Press, 2002: http://diwww.epfl.ch/~gerstner/BUCH.html
- 13. Ітада A. A random behavior of spiking neurons: Towards associative memory. // Нейрокомпьютеры: разработка и применение. -2004. -N1. -c. 19-22.
- 14. Jaeger H. The "echo state" approach to analyzing and training recurrent neural networks. // GMD Report 148, German National Research Center for Information Technology. -2002.
- 15. Hebb, D.O. The Organization of behavior. New York, 1949.
- 16. Hochreiter S., Schmidhuber J. Long Short-Term Memory. // Neural Computation. 1997. vol. 8. pp. 1735-180.
- 17. Hoppensteadt F. C., Izhikevich E. M. Synchronization of Laser Oscillators, Associative Memory, and Optical Neurocomputing. // Physical Review E. -2000.- vol. 62.- pp. 4010-4013.
- 18. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities. // Proceedings of the National Academy of Sciences. -1982.-vol. 79. -pp. 2554-2558.
- 19. Legenstein R., Maass W. What makes a dynamical system computationally powerful? // New Directions in Statistical Signal Processing: From Systems to Brain. MIT Press. -2005.
- 20. Maass W., Natschläger T. Emulation of Hopfield networks with spiking neurons in temporal coding. // Computational Neuroscience: Trends in Research. 1998. pp. 221-226.
- 21. Maass W., Natschläger T., Markram H. Real-time computing without stable states: A new framework for neural computations based on perturbations. // Neural Computation. -2002. vol. 11, pp. 2531-2560.
- 22. Markram H., Wang Y., Tsodyks M. Differential signaling via the same axon of neocortical pyramidal neurons. // Proceedings of the National Academy of Sciences. $-1998.-vol.\ 95.-pp.\ 5323-5328.$
- 23. Mérida-Casermeiroa E., Munõz-Pérez J. MREM: An associative autonomous recurrent network. // Journal of Intelligent & Fuzzy Systems. 2002. vol. 12. pp. 163-173.
- 24. Pantic L., Torres J. J., Kappen H. Associative memory with dynamical synapses. // Neural Computation. 2002. vol. 14. pp. 2903-2923.
- 25. Samengo I. Statistical description of associative memory. // AIP Conference Proceedings. -2003. vol. 658, issue 1. pp. 301-313.

О мощности ребер гиперграфа

Будников Ю. А., E-mail: y.budnikov@mail.ru

механико-математический факультет $M\Gamma Y$ г. Москва, ул. Саморы Машела, д.8, к. 1, кв. 99

Конечная проективная плоскость

Рассмотрим конечную проективную плоскость порядка q, где q - простое число. Она определяется так:

- 1) любые 2 точки лежат только на одной прямой;
- 2) любая прямая содержит ровно q+1 точку;
- 3)любая точка инциндентна ровно q+1 прямой;
- 4)любые 2 прямые пересекаются ровно в одной точке.

Далее сформулируем известную теорему в теории гиперграфов:

Teopeмa (Nicholas Pippenger and Joel Spencer)

Пусть $k \geq 2$ - фиксированное натуральное число. Будем рассматривать гиперграфы G = (V, E) - k -универсальные, т.е. каждое ребро которых содержит в точности k вершин(причем это число зависит от n(G) -числа вершин в графе G), и не тривиальные, т.е. содержащие хотя бы одно ребро. Будем для краткости называть их просто графами. Если G = (V, E), то V -множество вершин графа, E -множество его ребер. n(G) = |V(G)| -число его вершин. m(G) = |E(G)| -число ребер G.

Определим для любой вершины v $deg_G(v)$ —степень вершины v. $codeg_G(v,w)$ —костепень, т.е. число ребер, проходящее одновременно через вершины v,w. Обозначим расстояние между двумя ребрами графа G $\Delta(A,B)=$ минимальный номер $j\geqslant 0$: существует последовательность ребер H_0,\ldots,H_j графа $G:H_0=A,H_j=B,H_{f-1}\cap H_f\neq\emptyset$ для $1\leqslant f\leqslant j$.

Введем следующие обозначения.

$$D(G) = \max_{v \in V(G)} (deg_G(v)), \quad d(G) = \min_{v \in V(G)} (deg_G(v)), \quad C(G) = \max_{u, v \in V(G), u \neq w} (codeg_G(v, w))$$

Упаковка в графе G—набор P непересекающихся ребер G. Покрытие графа G - набор K ребер графа, содержащий все вершины графа. $\chi(G)$ —"хроматический индекс"графа G - минимальное число упаковок, на которые можно разбить все ребра G.

Теорема.

Для $\forall \delta > 0, \ k \geq 2$ - фиксированное число вершин на ребре. $\exists \delta' > 0, \ N > 0$, : из условий

$$d(G) \geqslant (1 - \delta')D(G) \tag{1}$$

$$C(G) \leqslant \delta' D(G) \tag{2}$$

для
$$n > N$$
 следует $\chi(G) \leqslant (1+\delta)D(G)$ (*)

Данную теорему можно переформулировать в терминах пределов, введя обозначения:

 $f(G) \sim g(G)$ для $\lim_G f(G)/g(G) = 1$, имеется в виду, что $n(G) \to \infty$

 $f(G) \prec g(G)$ для $\lim_G f(G)/g(G) = 0$, $f(G) \lesssim g(G)$ для $\lim_G f(G)/g(G) \leqslant 1$

Поэтому ввиду неравенств $d(G) \leq D(G) \leq \chi(G)$ можно переписать теорему так:

$$d(G) \sim D(G) \tag{1}$$

$$C(G) \prec D(G)$$
 (2)

 \Rightarrow

$$\chi(G) \sim D(G) \tag{*}$$

Один пример

Определение. Пусть некоторая функция k(n) возрастает по n в смысле (!).

1)если
$$N_1 < N_2$$
, где $N_1 > 0, N_2 > 0, N_1 \in \mathbb{N} < N_2 \in \mathbb{N} \Rightarrow k(N_1) \le k(N_2)$
2) $\forall N > 0: N \in \mathbb{N} \Rightarrow \exists \ \tilde{N} > N: k(\tilde{N}) > k(N)$

В тех же обозначениях теперь будем рассматривать число вершин на ребре k как возрастающую (в смысле (!)) функцию от n - числа вершин гиперграфа: $k(n) \to \infty$, при $n \to \infty$.

Везде далее рассматриваем возрастающие функции в смысле (!). Условие (2) в определении существенно, т.к. для остальных функций подходит теорема (Nicholas Pippenger and Joel Spencer). Условие (1) существенно, т.к. иначе $k(n) \equiv n$.

Покажем, что в предыдущей теореме нельзя запустить рост числа вершин на ребре, т.е. докажем следующее утверждение.

 $\forall \delta$ $\forall \, \delta'$ 0, Утверждение. Для 0, любой возрастаюдля в смысле (!) ϕ ункции g(n)можно nocmpoumbвозрастающую k(n)число вершин на ребре - $ma\kappa$, чтоk(n)< $g(n), \forall n$ \in класс гиперграфов G(n), где n - число вершин гиперграфа c ростом максимальной степени D(n) : nри $n=n_k, k=1,2...$ - возрастающая подпоследовательность натуральных чисел - будет

$$d(G) \geqslant (1 - \delta')D(G) \tag{1}$$

$$C(G) \leqslant \delta' D(G) \tag{2}$$

но
$$\chi(G(n)) = k(n)(D(G(n)) - 1) + 1, \forall n \in \mathbb{N}$$
 (*)

Построим пример класса, подтверждающий это.

Пусть P(q) - проективная плоскость порядка q. Естественно, что k=q+1, всего точек в этой проективной плоскости $q^2+q+1=:n_0$. Запускаем рост числа точек на ребре $k=k(n_0), n_0\to\infty$. Это некая возрастающая функция от n_0 порядка $O(\sqrt{n_0})$. Наберем таких плоскостей при каждом фиксированном n_0 достаточное число $n=l(n_0)*n_0$: плоскости не пересекаются друг с другом и получается нужный фиксированный рост $k(n), n\to\infty$. Поскольку $l(n_0)\in\mathbb{N}$ - выбирается произвольно, то, очевидно, получим нужный рост $k(n), n\to\infty$, выбрав соответствующие $l(n_0)$ при $n_0=q^2+q+1$, где q - некоторое простое число(достаточно большое, выбирающееся на каждом шаге для увеличения n_0). Как раз такую подпоследовательность n и берем в качестве n_k из утверждения 1.

Поскольку при таком добавлении проективных плоскостей не изменяется длина ребра, степени вершин, костепени, а меняется только общее число вершин n, то утверждение 1 доказано, поскольку:

- 1) таким способом можно построить k(n) любого порядка малости относительно n, определенную на соответствующей подпоследовательности $n=n_k, k=1,2...;$
- 2) любой вершины степень же является возрастаn, тождественно k(n)любом функцией равной при из подпоследовательности $n = n_k, k = 1, 2...$ Поэтому выполнено (1);
- 3) костепень любой вершины равна 1 при любом n, что непосредственно следует из определения проективной плоскости. Поэтому выполнено (2);
- 4) $\chi(G) = k(n) * (D(n) 1) + 1$ при каждом фиксированном n, т.е. максимально возможному числу(как показано в Утверждении (Nicholas Pippenger and Joel Spencer), приведенном в их статье). Поэтому выполнено (*);

Замечание При каждом фиксированном n для некоторой подпоследовательности $n=n_k, k=1,2...$ гиперграф G можно сделать связным. Берем дополнительное ребро и связываем им первые два несвязанных графа из $l(n_0)$ штук: $\left[\frac{k(n_0)}{2}\right]$ точек принадлежат первому графу, а оставшиеся второму. Далее аналогично связываем второй и третий графы и т.д. При этом степень каждого графа на n_0 вершинах в связке увеличивается максимум на 2, костепень - на $2, \chi(G)$ не уменьшается. Поэтому (1) и (2), (*) будут так же выполнены для такого класса уже связных гиперграфов на n вершинах для некоторой подпоследовательности нашей последовательности $n=n_k, k=1,2...$, при $n_k>N(\delta,\delta')$.

Список литературы

[1.] Asymptotic behavior of the Chromatic Index for Hypergraphs, Nicholas Pippenger and Joel Spencer, Journal of combinatorial theory, Series A 51, 24-42(1989).

Теоретико-множественные конструкции полного образа, ограничения, конфинальности и совместности в основаниях реляционных баз данных

Буй Дмитрий Борисович

Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, доктор физико-математических наук, зав. лабораторией: 03022, Украина, Киев, пр. Глушкова 2, корп. 6; e-mail: buv@unicvb.kiev.ua

Кахута Надежда Дмитриевна

Киевский механико-муталлургический техникум, зам. директора по учебной работе, преподаватель математики: 02090, Украина, Киев, Харковское шоссе 15; e-mail: vkahuta@iba.com.ua.

Приводятся общезначимые свойства (монотоность, дистрибутивность, убываемость, сохранение коммутативнсти и ассоциативности при расширении бинарних функций на множества и т.д.) теоретико-множественных конструкций полного образа, ограничения, теоретико-множественных отношений конфинальности множеств и совместности отношений, задана структура множества частичных функций, упорядоченных по включению графиков. Выбор указанных конструкций обусловлен естественными представлениями через них сигнатурных операций табличных алгебр.

Ключевые слова: бинарное отношение, полный образ, ограничение, конфинальность, совместность, частичные функции.

Доклад посвящен общим свойствам теоретико-множественных конструкций: (1) полного образа множества относительно бинарного отношения, (2) ограничения бинарного отношения по множеству, (3) отношения конфинальности, (4) отношения совместности, (5) устройству множества частичных функций, упорядоченных по включению графиков.

Выбор именно этих конструкций обусловлен, во-первых, их общезначимостью и, во-вторых, естественным применением в теории реляционных баз данных при исследовании табличных алгебр, построенных на основе известных реляционных алгебр Кодда [1-3; 4, подразд. 1.3, с. 22-24]. Дело в том, что сигнатурные операции табличных алгебр имеют простые представления через указанные конструкции, что позволяет переносить свойства конструкций на табличные операции.

Зафиксируем универсум D, элементы которого обозначим x, y, z,.... Подмножества универсума обозначим X, Y, ..., бинарные отношения на D-U, V,...

Как обычно, бинарное отношение U назовем ϕ ункциональным, если для всех x, y, z выполняется < x, $y > \in U \& < x$, $z > \in U \implies y = z$. Функциональные отношения (частичные ϕ ункции) обозначим f, g, Область определенности ϕ ункции f обозначим domf, она совпадает с проекцией ϕ ункции по первой компоненте – $domf = \pi_1^2 f$, где $\pi_i^2 U$ – проекция бинарного отношения по i-ю компоненте, i=1,2.

Бинарное отношение U назовем *инъективным*, если для всех x, y, z выполняется < x, $y > \in U \& < z$, $y > \in U \implies x = z$. Очевидно, что имеет место: U – функционально $\iff U^{-1}$ инъективно;

здесь
$$U^{-1} \stackrel{def}{=} \{ < x,y> \mid < y,x> \in U \}$$
 – отношение, обратное к U .

Полный образ.

Как обычно, nолным образом множества X относительно отношения U называется множество def

 $U[X] = \{ y \mid \exists x (x \in X \& < x, y > \in U) \};$ композицией отношений U и V называется отношение def

 $def \\ U\circ V=\{< x,y>\mid \exists z(< x,z>\in V\&< z,y>\in U)\}. \ \ \it{Пустое отношение}\ \ o$ бозначим $\ \varepsilon$. Отношение $\ U$ назовем $\it{momaльным}$, если $\ \pi_1^2U=D$.

Предложение 1. Для полного образа выполняется:

1.
$$U_1 \subseteq U_2 \& X_1 \subseteq X_2 \Rightarrow U_1[X_1] \subseteq U_2[X_2]$$
 (монотонность);

2.
$$U[\bigcup_i X_i] = \bigcup_i U[X_i], (\bigcup_i U_i)[X] = \bigcup_i U_i[X]$$
 (дистрибутивность относительно объединений);

3.
$$U[\bigcap X_i] \subseteq \bigcap U[X_i]$$
 (верхняя оценка полного образа пересечения);

4.
$$U_1[U_2[X]] = (U_1 \circ U_2)[X]$$
 (полный образ относительно композиции отношений);

5.
$$U[X] = U[X \cap \pi_1^2 U];$$

6. $U[X] = \emptyset \Leftrightarrow X \cap \pi_1^2 U = \emptyset$ (критерий пустоты полного образа); в частности, $\varepsilon[X] = U[\emptyset] = \emptyset$ (сохранение пустого отношения и пустого множества) и, в предположении тотальности отношения $U, U[X] = \emptyset \Leftrightarrow X = \emptyset$;

7.
$$U[X]\setminus U[Y]\subseteq U[X\setminus Y]\subseteq U[X]$$
 (нижняя и верхняя оценка полного образа разности). \square

Пп. 3 і 7 (нижняя оценка) ставят вопрос о естественных достаточных условиях для выполнения соответствующих равенств. Ниже, как обычно, *ограничением отношения U по множеству X* называется def отношение $U \mid X = U \cap (X \times D) = U \cap (X \times \pi_2^2 U)$.

Предложение 2. Для полного образа выполняется:

1.
$$U \mid \bigcup_{i \in I} X_i$$
 инъективно $\Rightarrow U[\bigcap_{i \in I} X_i] = \bigcap_{i \in I} U[X_i];$

2.
$$f^{-1}[\bigcap_{i} X_{i}] = \bigcap_{i} f^{-1}[X_{i}];$$

3.
$$U \mid (X \cup Y)$$
 инъективно $\Rightarrow U[X \setminus Y] = U[X] \setminus U[Y];$

4.
$$f^{-1}[X \setminus Y] = f^{-1}[X] \setminus f^{-1}[Y]$$
. \Box

Как показывают простые примеры, условия предложения 2 для отношений являются достаточными, но не необходимыми.

Расширение отношений на множества.

Полный образ естественно позволяет по операциям на универсуме строить операции на его булеане. Через [f] обозначим унарную тотальную операцию на булеане универсума D, которая индуцируется def

частичной функцией f і задается равенством [f](X) = f[X]. Пусть F — бинарная частичная операция на D; она также индуцирует бинарную тотальную операцию [F] на булеане D — def $[F](X,Y) = F[X \times Y]$.

Частичную бинарную операцию F назовем коммутативной (ассоциативной), если выполняется обобщенное равенство $F(x,y) \cong F(y,x)$ для всех x,y (соответственно $F(F(x,y),z) \cong F(x,F(y,z))$ для всех x,y,z). Здесь под обобщенным равенством понимается равенство, в котором обе части или не определены, или определены и имеют одинаковые значения. Связь между свойствами исходных и индуцированных операций раскрывают два следующих предложения.

Предложение 3. Функция f инъективна и тотальна \Leftrightarrow операция [f] инъективна. \square

Очевидно, что тотальность функции f существенна для необходимости; вместе с тем, переходя к отношениям, необходимость обобщается в виде импликации: отношение U инъективно и тотально \Longrightarrow операция [U] инъективна.

Выше отношение расширяется на булеан по той же схеме: [U](X) = U[X]. Вместе с тем, простые примеры для счетного универсума показывают, что функциональность существенна для достаточности.

Предложение 4. Частичная операция F коммутативна (ассоциативна) \Leftrightarrow тотальная операция [F] коммутативна (ассоциативна). \square

Ограничение.

Параметрический оператор $U \mapsto U \mid X$ обозначим $\uparrow X$. Ниже, как обычно, под *оператором замыкания* на частично упорядоченном множестве понимается идемпотентный, монотонный и убывающий (либо возрастающий) оператор.

Предложение 5. Для ограничения выполняется:

1.
$$U_1 \subseteq U_2 \& X_1 \subseteq X_2 \Rightarrow U_1 \mid X_1 \subseteq U_2 \mid X_2$$
 (монотонность ограничения и оператора $\uparrow X$);

2.
$$\pi_1^2(U \mid X) = \pi_1^2 U \cap X$$
 (проекция ограничения);

3. $U \mid X = \varepsilon \Leftrightarrow \pi_1^2 U \cap X = \emptyset$ (критерий пустоты ограничения); в частности, $\varepsilon \mid X = U \mid \emptyset = \varepsilon$ (сохранение пустого отношения и пустого множества);

4.
$$U \mid X = U \mid (X \cap \pi_1^2 U), U = U \mid \pi_1^2 U;$$
 в частности, $\pi_1^2 U \subseteq X \Leftrightarrow U \mid X = U;$

5. $(U \mid X) \mid Y = U \mid (X \cap Y)$, или в операторном виде $\uparrow Y \circ \uparrow X = \uparrow (X \cap Y)$; в частности, $\uparrow X \circ \uparrow X = \uparrow X$ (идемпотентность оператора $\uparrow X$);

6.
$$U \mid X \subseteq U$$
 (убываемость оператора $\uparrow X$);

7. оператор $\uparrow X$ является оператором замыкания относительно теоретико-множественного включения \subset ;

8.
$$(\bigcup_i U_i) \mid X = \bigcup_i U_i \mid X, U \mid \bigcup_i X_i = \bigcup_i U \mid X_i$$
, $(\bigcap_i U_i) \mid X = \bigcap_i U_i \mid X, U \mid \bigcap_i X_i = \bigcap_i U \mid X_i$ (дистрибутивность ограничения относительно объединений и пересечений);

9. $f\subseteq g \& X\subseteq domf\Rightarrow f\mid X=g\mid X;$ в частности, $f\subseteq g\Rightarrow f=g\mid domf$, $f\subseteq g \& domf=domg\Rightarrow f=g;$

10.
$$(U \mid X)[Y] = U[X \cap Y]$$
 (полный образ множества относительно ограничения). \Box

Бинарное отношение конфинальности множеств.

Зафиксируем частичный порядок \leq на универсуме D. Этот порядок индуцирует обычное бинарное def отношение конфинальности \prec на булеане универсума: $X \prec Y \Leftrightarrow \forall x (x \in X \Rightarrow \exists y (y \in Y \& x \leq y))$.

Введенное отношение рефлексивно, транзитивно, имеет наименьшим элементом пустое множество \emptyset . Т.е. для того, чтобы быть частичным порядком, не хватает антисимметричности, поэтому рассмотрим ограничение этого отношения.

Множество X назовем $\partial ucкретным$, если частично упорядоченное множество $< X, \le >$ дискретно в стандартном смысле: для всех $x_1, x_2 \in X$ выполняется $x_1 \le x_2 \Longrightarrow x_1 = x_2$.

Предложение 6. Отношение конфинальности частично упорядочивает семейство дискретных множеств.

Отметим, что два отношения на таблицах, которые рассматриваются в [2; 4, подразделы 2.6, 2.8], строятся именно по указанной схеме.

Приведем общие свойства отношения конфинальности, обозначая верхний (нижний) конус множества X как X^{Δ} (соответственно X^{∇}), а супремум (инфимум) этого же множества как $\coprod X$ (соответственно $\prod X$).

Предложение 7. Для произвольных множеств выполняется $X \prec Y \Rightarrow Y^{\Delta} \subseteq X^{\Delta}$. Для произвольных множеств при условии существования их супремумов выполняется $X \prec Y \Rightarrow \coprod X \leq \coprod Y$. Если Y конфинальное подмножество множества X, то $X^{\Delta} = Y^{\Delta}$; т.е. $\coprod X \simeq \coprod Y$.

Таким образом, вопрос о супремуме множества эквивалентен этому же вопросу для его конфинального подмножества. Согласно принципу двойственности при переходе к обратному порядку все утверждения для конфинальности переходят в утверждения для коннициальности.

Совместность, устройство множества частичных функций.

Далее под функциями понимаем частичные функции. Введем бинарное *отношение совместности отношений*:

$$U pprox V \Leftrightarrow U \mid X = V \mid X,$$
 где $X = \pi_1^2 U \cap \pi_1^2 V$.

Очевидно, что это отношение рефлексивно, симметрично, но не транзитивно. Как обычно, множество функций F назовем совместным, если для любого конечного подмножества $\{f_1,...,f_n\}\subseteq F$ найдется функция h, такая, что $f_1,...,f_n\subseteq h$. Ниже пустое отношение ε обозначим f_{\emptyset} (всюду неопределенная функция), множество всех (частичных) функций на универсуме обозначим F. В литературе, как правило, ограничиваются индуктивностью частично упорядоченного множества (ч. у. м.) $<\mathbf{F}, \subseteq >$; на самом деле о структуре этого ч. у. м. можно сказать гораздо больше.

Предложение 8. Выполняются следующие утверждения:

 $1.\ f_{\varnothing}$ – наименьший элемент ч. у. м. $<\mathbf{F},\subseteq>$, которое имеет наибольший элемент тогда и только тогда, когда универсум D имеет не более одного элемента;

2.
$$\prod F = \bigcap_{f \in F} f$$
 для любого непустого множества функций $F \subseteq \mathbf{F}$;

3. следующие 5 утверждений эквивалентны для любого множества функций F:(1) супремум $\coprod F$ существует; (2) множество F ограничено сверху; (3) множество F совместно; (4) все функции множества F попарно совместны - $\forall f \forall g (f,g \in F \Rightarrow f \approx g);$ (5) отношение $\bigcup_{f \in F} f$ функционально.

Таким образом, множество частичных функций когерентно (супремумы совместных множеств существуют) и является полной полурешеткой (существуют супремумы ограниченных сверху множеств). Это факт не случаен, а является следствием совпадения понятий когерентного множества и полной полурешетки в классе индуктивных множеств.

- 1. Редько В. Н., Буй Д. Б. К основаниям теории реляционных моделей баз данных // Кибернетика и системный анализ. -1996. -№4. С. 3-13.
- 2. Редько В. Н., Брона Ю. И., Буй Д. Б. Реляционные алгебры: операции проекции и соединения // Кибернетика и системный анализ. − 1997. − №4. − С. 89-100.

- 3. Редько В. Н., Брона Ю. И., Буй Д. Б. Реляционные алгебры: операции деления и переименования // Кибернетика и системный анализ. − 1997. − №5. − С. 3-15.
- 4. Редько В. Н., Брона Ю. Й., Буй Д. Б., Поляков С. А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. Київ: Видавничий дім "Академперіодика", 2001. 198 с.

Формализация структурных ограничений связей в модели "сущность-связь"

Буй Дмитрий Борисович

Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, доктор физико-математических наук, зав. Лабораторией,

03022, Украина, Киев, пр. Глушкова 2, корп. 6;

e-mail: <u>buy@unicyb.kiev.ua</u>

Сильвейструк Людмила Николаевна

Киевский национальный университет имени Тараса Шевченко, факультет кибернетики, 03022, Украина, Киев, пр. Глушкова 2, корп. 6; e-mail: slm-klm@rambler.ru

Рассматриваются и формализуются в терминах теории отношений такие основные понятия модели "сущность-связь": сущности, связи, структурные ограничения связей (показатель кардинальности, степень участия, структурные ограничения вида (min, max)). Для бинарных отношений введены два оператора min, max, в терминах которых задаются указанные структурные ограничения; приведена основная теорема о совместности значений этих операторов на исходном отношении и отношении, обратном к нему.

Ключевые слова: сущность, связь, показатель кардинальности, степень участия, структурное ограничение вида (min, max).

Существуют разные подходы к моделированию данных, каждый из которых имеет своих поклонников. Одна из таких моделей – ER-модель (Entity-Relationship model, модель "сущность-связь") [1]. Эта модель стала традиционной и наиболее популярной. В данной работе формализуются некоторые из основных понятий модели.

Сущности и связи. Тип сущности будем интерпретировать как множество, а сущность – как элемент этого множества. Типы связей будем уточнять в виде (конечноарных) логико-математических отношений; в частности, типы бинарных связей – в виде бинарных отношений.

Показатель кардинальности. В модели существуют стандартные ограничения, которые накладываются на типы связи. Одно из таких ограничений — показатель кардинальности. Среди бинарных типов связи выделяют типы связи с показателями кардинальности "один к одному" (1:1), "многие к одному" (M:1), "один ко многим" (1:M) и "многие ко многим" (M:N).

Допустим, что R — тип связи, который соединяет типы сущностей E и F. Для адекватной формализации показателя кардинальности типы сущностей E и F будем интерпретировать как множества E и F соответственно, а тип связи R — как бинарное отношение R, причем $R \subseteq E \times F$. Как обычно $R^{-1} \subset F \times E$ — отношение, обратное к R.

В табл. 1 показана взаимосвязь между показателями кардинальности типа связи и свойствами функциональности бинарных отношений \mathbf{R} , \mathbf{R}^{-1} . Можно сделать вывод: ограничение "один ко многим" ("многие к одному") связано с функциональностью (эквивалентно: с инъективностью); "один к одному" – с одновременной функциональностью и инъективностью; наконец "многие ко многим" – с бинарными отношениями общего вида. Здесь нужно учесть очевидную логическую связь между функциональностью и инъективностью: бинарное отношение функциональное тогда и только тогда, когда обратное отношение инъективное (см., например [2, утверждение 1]).

Табл. 1 — Взаимосвязь между показателем кардинальности и функциональностью бинарных отношений, а также значениями оператора max

		Отношение R	
		R функциональное	R не функциональное
	R ⁻¹ функ- циональное	R – "один к одному" (1:1)	R — "многие к одному" (M:1), который направлен от F к E R — "один ко многим" (1:M), который направлен от E к F
Отн о-шение		$\max(R) \le 1 \ \land \ \max(R^{-1}) \le 1$	$2 \le \max(R) \le \infty \land \max(R^{-1}) \le 1$
\mathbf{R}^{-1}	R ⁻¹ не функциональное	R — "многие к одному" (М:1), который направлен от E к F R — "один ко многим" (1:М), который направлен от F к E	R – "многие ко многим" (M:N)
		$\max(R) \le 1 \ \land \ 2 \le \max(R^{-1}) \le \infty$	$2 \le \max(R) \le \infty \ \land \ 2 \le \max(R^{-1}) \le \infty$

Степень участия сущности в связи. Существует еще одно ограничение для типов связи — *степень участия сущности в связи.* Различают два вида участия типа сущности в типе связи: полное и частичное.

Формализовать понятие полного и частичного участия можно посредством проекции отношения (табл. 2). В предыдущих обозначениях имеем следующую таблицу, где $\pi_1^2(\mathbf{R})$, $\pi_2^2(\mathbf{R})$ проекции бинарного отношения соответственно по первой и второй компонентам.

Табл. 2 – Взаимосвязь между степенью участия и проекцией отношения, а также значениями оператора min

Проекция отношения	Степень участия типа сущности в связи	Значение оператора min
$\pi_1^2(\mathbf{R}) = \mathbf{E}$	участие типа сущности Е в типе связи R полное	$0 < \min(\mathbf{R})$
$\pi_1^2(\mathbf{R}) \subset \mathbf{E}$	участие типа сущности Е в типе связи R частичное	$\min(\mathbf{R}) = 0$
$\pi_2^2(\mathbf{R}) = \mathbf{F}$	участие типа сущности F в типе связи R полное	$0 < \min(\mathbf{R}^{-1})$
$\pi_2^2(\mathbf{R}) \subset \mathbf{F}$	участие типа сущности F в типе связи R частичное	$\min(\mathbf{R}^{-1}) = 0$

Структурные ограничения вида (темп. так). Существует и альтернативный вариант рассмотрения ограничений на типах связи — так называемые структурные ограничения, которые предусматривают задание минимальных и максимальных значений (темп. так).

Уточнить такие структурные ограничения можно посредством понятия полного образа. Как и ранее типы сущностей E, F будем интерпретировать как непустые множества E, F; элементы которых будем обозначать как x, y,...

Пусть $x \in \mathbf{E}$, полный образ одноэлементного множества $\{x\}$ относительно отношения \mathbf{R} oбозначим через R[x] ($R[x] = \{y \mid y \in F \land \langle x, y \rangle \in R\}$). Будем считать, что множества \mathbf{E} , \mathbf{F} не более чем счетные, а все полные образы одноэлементных множеств конечные.

Множество мощностей полных образов всех элементов множества ${\bf E}$ обозначим как ${\bf Im}({\bf R})$, т.е.

 $Im(\mathbf{R}) = \{ |\mathbf{R}[x]| | x \in \mathbf{E} \}$. Очевидно, что $Im(\mathbf{R})$ — непустое подмножество натуральных чисел, конечное или бесконечное. В любом случае это множество имеет наименьший элемент, который обозначим, как $min(\mathbf{R})$. Наибольшего же элемента множество $Im(\mathbf{R})$ может и не иметь, поэтому введем следующее обозначение:

 $\max(\!R\!) = \begin{cases} \text{наибольший} елементмножества \! Im(\!R\!), & \textit{если } Im(\!R\!) \, \textit{конечноемножество} \\ \infty, & \textit{в противномслучае} \end{cases}$

По существу множество натуральный чисел N со стандартным порядком \leq пополнили def наибольшим элементом ∞ , превратив его в полную решетку $< N', \leq >$, где $N' = N \cup \{\infty\}$, причем $n < \infty$ для всех $n \in N$. Непосредственно из определений вытекают равенства $\min(R) = \prod \operatorname{Im}(R)$, $\max(R) = \coprod \operatorname{Im}(R)$, где символы \prod , \coprod используются для обозначения инфимумов и супремумов соответственно (в полной решетке N').

В диаграммах модели «сущность-связь» структурные ограничение min, max указываются по обе стороны связи, соединяющей типы сущностей. Возникает вопрос о существовании логической связи между указанными значениями. Переходя к точным терминам, требуется установить логическую взаимосвязь между значениями введенных операторов на исходном отношении (\mathbf{R}) и на отношении, обратном к нему (\mathbf{R}^{-1}). Эта задача будет решена в приведенной далее теореме, доказательство которой опирается на следующие леммы об основных свойствах операторов min, max .

Лемма 1. Для произвольного бинарного отношения ${f R}$ выполняются следующие утверждения:

- 1. $\min(R) \le \max(R)$, bose more $\max(R) = \infty \Rightarrow \min(R) < \max(R)$;
- 2. $\min(R) = \max(R) \Leftrightarrow \forall x \ y \ (x, y \in E \Rightarrow |R[x]| = |R[y]|);$
- 3. $R = \emptyset \Leftrightarrow \min(R) = \max(R) = 0$; более того $R = \emptyset \Leftrightarrow \min(R) = \max(R) = \min(R^{-1}) = \max(R^{-1}) = 0$ (характеристическое свойство пустого бинарного отношения);
 - 4. $\pi_1^2(R) \subset E \Leftrightarrow \min(R) = 0$, $\pi_1^2(R) = E \Leftrightarrow \min(R) > 0$;
 - 5. $0 < \min(R) < \max(R) \Rightarrow \left| \pi_1^2(R) \right| \ge 2$;
 - 6. $\max(R) = \infty \Rightarrow \left|\pi_1^2(R)\right| = \left|\pi_2^2(R)\right| = \omega$, где \mathcal{O} кардинал счетных множеств;
 - 7. **R** функциональное отношение \iff $\max(R) \le 1$.

Из пунктов 4, 7 этой леммы вытекает заполнение таблиц 1, 2 применительно к операторам min, max.

Лемма 2. Пусть $|\mathbf{E}| = l > 0$, $|\mathbf{F}| = k > 0$, $a \ U(l,k) = \mathbf{E} \times \mathbf{F}$ — универсальное отношение на множествах \mathbf{E} , \mathbf{F} ; тогда $\min(\mathbf{R}) = \max(\mathbf{R}) = l \ u \ \min(\mathbf{R}^{-1}) = \max(\mathbf{R}^{-1}) = k$.

Значения операторов \min, \max зависят не только от аргумента-отношения (в предыдущих обозначениях R), но и от множества-параметра, которому принадлежат первые компоненты пар отношения (множества E); поэтому точнее было бы писать, например, $\min_E(R)$ вместо $\min(R)$.

Лемма 3. Пусть отношение \mathbf{R} и множества \mathbf{E} , \mathbf{F} , \mathbf{E}' , такие, что $\mathbf{R} \subseteq \mathbf{E} \times \mathbf{F}$ и $\mathbf{E} \subset \mathbf{E}'$; тогда $\min_{\mathbf{E}'}(\mathbf{R}) = 0$ и $\max_{\mathbf{E}}(\mathbf{R}) = \max_{\mathbf{E}'}(\mathbf{R})$.

Следующая лемма рассматривает случай, когда отношение имеет структуру объединения попарно совместных отношений (совместность в понимании [3], т.е. $U \approx V \Leftrightarrow U \mid \mathbf{X} = V \mid \mathbf{X}$, где, например, $U \mid \mathbf{X}$ — ограничение отношения U по множеству \mathbf{X}), в лемме выражаются значения операторов min, тах на исходном множестве через значение тех же операторов на множествах из объединения.

Лемма 4. Пусть отношение R такое, что $R = \bigcup_{i \in I} R_i$, причем все отношения R_i , $i \in I$ попарно совместные. Тогда $\max_E(R) = \coprod_{i \in I} \max_E(R_i)$, где множества E, E_i , такие, что $\pi_1^2(R) \subseteq E$, $\pi_1^2(R_i) \subseteq E_i$ для всех $i \in I$. Кроме того, обозначая проекции по первой компоненте отношений R и R_i через G и G_i соответственно, имеем равенство $\min_G(R) = \prod_{i \in I} \min_{G_i}(R_i)$.

Теорема. Для ячеек табл. 3, обозначенных знаком «+», существуют отношения с соответствующими значениями min, max . Для ячеек табл. 3, обозначенных знаком «—», не существуют отношения с соответствующими значениями min, max .

Табл. 3 – Все варианты значений min, max и их совместимость для отношения и обратного к нему отношения

		$l' = \min(\mathbf{R}^{-1}), \ l = \max(\mathbf{R}^{-1})$					
		l' = l = 0	<i>l'</i> = <i>l</i> > 0	$l'=0, l\geq 1$	$l'=0, l=\infty$	$l' \geq 1, l \geq l'$	$l' \ge 1, l = \infty$
	k' = k = 0	+	-	-	-	-	_
$k' = \min(\mathbf{R})$ def $k = \max(\mathbf{R})$	k' = k > 0	-	+	+	+	+	+
	$k'=0, k\geq 1$	_	+	+	+	+	+
	$k'=0, k=\infty$	_	+	+	+	+	+
	$k' \ge 1, k \ge k'$	_	+	+	+	+	+
	$k' \ge 1, k = \infty$	_	+	+	+	+	+

Из этой теоремы следует, что, за исключением тривиального случая пустого отношения, для любого распределения значений операторов min, max существует отношение, на котором указанные значения достигаются.

После рассмотрения ограничений на типы связей (табл. 1-2) можно сделать вывод: структурное ограничение вида (min, max) является более выразительным, чем показатель кардинальности и степень участия сущности в связи.

- 1. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: пер. с англ. Москва: "Вильямс", 2000.– 1120 с.
- 2. Буй Д. Б., Кахута Н. Д. Властивості теоретико-множинних конструкцій повного образу та обмеження // Вісник Київського університету. Серія: фіз.-мат. науки. 2005. Вип. 2. С. 232-240.
- 3. Редько В. Н., Брона Ю. Й., Буй Д. Б., Поляков С. А. Реляційні бази даних: табличні алгебри та SQL-подібні мови. Київ: Видавничий дім "Академперіодика", 2001. 198 с.

Об автоматной модели преследования

Волков Н. Ю.,

кафедра МаТИС механико-математического факультета МГУ им. М. В. Ломоносова

Изучается процесс преследования системой автоматов ("хищников") нескольких независимых друг от друга автоматов ("жертв") на плоскости. Показано, что существует конечный коллектив хищников, который "ловит" любую конечную независимую систему жертв, таких, что их скорость меньше скорости хищников, и их обзор не больше обзора хищников, при любом начальном расположении жертв и стартующих из одной точки хищников.

Введение

Рассматривается автоматный аналог ситуации преследования хищниками своих жертв. С этой целью формализуются пространство, в котором происходит процесс преследования (пространство преследования), понятия хищников и жертв, как по логике их поведения, так и по способности обнаруживать друг друга и перемещаться. Пространство преследования представляется в виде плоскости, разбитой на квадраты целочисленной решеткой, а хищники и жертвы — в виде автоматов, которые, находясь в какой-либо клетке, умеют обозревать некоторую ее окрестность, и, в зависимости от вида (конфигурации) этой окрестности (т.е. от расположения других автоматов в этой окрестности) и своего состояния, способны перемещаться в другую клетку плоскости. Автоматы-хищники и автоматы-жертвы в начале процесса образуют определенную диспозицию, находясь в своих начальных состояниях. После этого начинается процесс перемещения автоматов по плоскости. Внутренние логики автоматов в совокупности определяют этот процесс. Жертва считается пойманной, если она оказалась в фиксированной окрестности одного из хищников.

Рассматриваются следующие 2 основные задачи.

Сначала предполагается, что каждый из хищников "не видит" других хищников, но видит жертв, попавших в его зону обзора. Аналогично, каждая жертва "не видит" других жертв, но видит хищников, попавших в ее зону обзора (в этом случае мы говорим, что хищники и жертвы представляют собой, соответственно, независимые системы автоматов). Спрашивается, существует ли система хищников K, такая что для любой системы жертв S, найдется такое начальное расположение автоматов из K, что при любом начальном расположении автоматов из S с течением времени все жертвы будут пойманы хищниками.

Затем рассматривается более сложный и более реальный случай, когда хищники "видят" и жертв и друг друга на расстоянии своего обзора (в этом случае мы говорим, что хищники представляют собой коллектив автоматов), а жертвы не обладают этим свойством (представляют собой независимую систему автоматов). Ставится тот же вопрос о возможности поимки всех жертв.

Показывается, что первая задача не имеет положительного решения, а вторая решается положительно. Устанавливается, что существует коллектив из 28 хищников, такой, что хищники, стартуя из начального расположения в одной произвольной клетке плоскости, ловят любую конечную независимую систему жертв при любом начальном расположении жертв.

Автор работы выражает признательность В.Б.Кудрявцеву за научное руководство.

Постановка задачи и основные результаты

Будем использовать стандартные обозначения для множеств натуральных и целых чисел $\mathbb N$ и $\mathbb Z$, соответственно. Положим $\mathbb N_0 = \mathbb N \cup \{0\}$. Множество клеток, на которые плоскость разбивается целочисленной решеткой, обозначим через $\mathbb Z^2$, сопоставляя каждой клетке координаты ее нижнего левого угла. Назовем r-окрестностью клетки (x_0,y_0) множество $D_{(x_0,y_0),r} = = \{(x,y) \mid (|x-x_0| + |y-y_0|) \le r\}$.

Под автоматом будем понимать инициальный конечный автомат вида $\mathcal{A}=(A,Q,B,\varphi,\psi,q_0)$, где A – входной, B – выходной, Q – внутренний алфавиты автомата $\mathcal{A},\ \varphi:Q\times A\to Q$ и $\psi:Q\times A\to B$ – функции переходов и выходов \mathcal{A} , соответственно, $q_0\in Q$ – его начальное состояние. Алфавит A определяет возможности \mathcal{A} "видеть" происходящее вокруг, а алфавит B – его возможности перемещаться. Алфавит Q и функции φ и ψ задают внутреннюю логику автомата \mathcal{A} .

Рассмотрим автомат \mathcal{A} , перемещающийся по \mathbb{Z}^2 . Выходным алфавитом \mathcal{A} является множество $B = D_{(0,0),V}$, где параметр $V \in \mathbb{N}$ называется *скоростью автомата* \mathcal{A} . Входной алфавит \mathcal{A} зависит

от параметра $R \in \mathbb{N}$ $(R \geq V)$, называемого *обзором автомата* \mathcal{A} и способа взаимодействия \mathcal{A} с другими автоматами. Возможны два случая такого взаимодействия:

- 1) ${\cal A}$ является элементом независимой системы автоматов;
- 2) A является элементом коллектива автоматов.

Автомат со скоростью V и обзором R будем обозначать как $\mathcal{A}(R,V)$. Пусть $\mathcal{A}(R,V)$ находится в клетке (x_0,y_0) . Множество $D_{(x_0,y_0),V}$ называется окрестностью хода \mathcal{A} , а множество $D_{(x_0,y_0),R}$ – зоной обзора \mathcal{A} .

Рассмотрим две системы автоматов $K = (W_1, \ldots, W_m)(R, V)$ (хищники) и $S = (U_1, \ldots, U_n)(R', V')$ (жертвы), где R и R' – обзоры, а V и V' – скорости хищников и жертв, соответственно. Здесь S – независимая система автоматов, K может быть независимой системой, а может при $m \geq 2$ быть коллективом.

Пусть хищники W_1, \ldots, W_m расположены в клетках $(x_1, y_1), \ldots, (x_m, y_m)$, а жертвы U_1, \ldots, U_n – в клетках $(x'_1, y'_1), \ldots, (x'_n, y'_n)$, соответственно.

Для каждого $i=1,\ldots,n$ строку (a_1,\ldots,a_m) , такую что для любого $j=1,\ldots,m$

$$a_j = \left\{ egin{array}{ll} (x_j - x_i', \ y_j - y_i') & , ext{если} \ |x_j - x_i'| + |y_j - y_i'| \leq R'; \ \Lambda & , ext{иначе}, \end{array}
ight.$$

назовем U_i -конфигурацией. Легко видеть, что $a_j \in (D_{(0,0),R'} \cup \{\Lambda\})$. Каждая U_i -конфигурация однозначно задает (кодирует) расположение хищников, находящихся в зоне обзора U_i , следующим образом. Если $a_j = \Lambda$, то W_j расположен вне зоны обзора U_i , а если $a_j = (x_j'', y_j'')$, то W_j расположен в клетке $(x_i' + x_j'', y_i' + y_j'')$.

Если $K = (\mathring{W}_1, \dots, \mathring{W}_m)$ – независимая система автоматов, то для каждого $j = 1, \dots, m$ строку (a_1, \dots, a_n) , такую что для любого $i = 1, \dots, n$

$$a_i = \left\{ \begin{array}{ll} (x_i' - x_j, \ y_i' - y_j) & \text{, если } |x_i' - x_j| + |y_i' - y_j| \leq R; \\ \Lambda & \text{, иначе,} \end{array} \right. \tag{*}$$

назовем W_j -конфигурацией. Легко видеть, что $a_i \in (D_{(0,0),R} \cup \{\Lambda\})$. Каждая W_j -конфигурация однозначно задает расположение жертв, находящихся в зоне обзора W_j .

Обозначим внутренний алфавит W_j как Q_j , а множество всех пар вида ((x,y),q), где $(x,y)\in D_{(0,0),R},\ q\in \bigcup_{j=1}^mQ_j$ – как M. Пусть каждый хищник W_j находится в состоянии q_j . Если $K=(W_1,\ldots,W_m)$ – коллектив автоматов, то для каждого $k=1,\ldots,m$ строку $(a_1,\ldots,a_n,a_{n+1},\ldots,a_{n+k+1},a_{n+k+1},\ldots,a_{n+m})$, такую что a_i при $i=1,\ldots,n$ определяется из (*), и для любого $j=1,\ldots,m,\ j\neq k$

$$a_{n+j} = \left\{ \begin{array}{ll} ((x_j-x_k,\,y_j-y_k),\,q_j\,) & \text{, если } |x_j-x_k|+|y_j-y_k| \leq R; \\ \Lambda & \text{, иначе,} \end{array} \right.$$

назовем W_j -конфигурацией. Легко видеть, что $a_{n+j} \in (M \cup \{\Lambda\})$. Каждая W_k -конфигурация однозначно задает расположение жертв в зоне обзора W_k , а также расположения и состояния других хищников, находящихся в зоне обзора W_k .

Расположения на плоскости жертв и хищников и состояния хищников однозначно задают все U_i -конфигурации и все W_j -конфигурации. Множество всех U_i -конфигураций при всевозможных расположениях жертв и хищников и состояниях хищников обозначим как F'. Аналогично, множество всех W_j -конфигураций обозначим как F. Входным алфавитом каждой жертвы U_i является множество всех пар вида $(\mathcal{F}_1, \mathcal{F}_2)$, где $\mathcal{F}_1 \in (\{\emptyset\} \cup F')$, а $\mathcal{F}_2 \in F'$. Входным алфавитом каждого хищника W_j является множество всех пар вида $(\mathcal{F}_1, \mathcal{F}_2)$, где $\mathcal{F}_1, \mathcal{F}_2 \in F$.

Момент времени 2τ ($\tau \in \mathbb{N}_0$) называется моментом хода жертв с номером τ . Момент ($2\tau + 1$) называется моментом хода хищников с номером τ . Промежуток времени [2τ , ($2\tau + 1$)] называется тактом с номером τ . Время взаимодействия автоматов будем измерять в тактах.

Преследование независимой системой \langle коллективом \rangle хищников независимой системы жертв происходит так. Фиксируются начальные (в нулевой момент времени) расположения всех хищников и жертв на плоскости. В нулевой момент каждая жертва U_i воспринимает в качестве входного символа пару $(\emptyset, \mathcal{F}_2)$, где U_i -конфигурация \mathcal{F}_2 задает расположение хищников в зоне обзора U_i . В момент 2τ $(\tau \in \mathbb{N})$ каждая жертва U_i воспринимает в качестве входного символа пару $(\mathcal{F}_1, \mathcal{F}_2)$, задающую расположения хищников в зоне обзора U_i в моменты $(2\tau - 1)$ и 2τ . В каждый момент

 2τ $(\tau \in \mathbb{N}_0)$ жертва U_i , в соответствии со своими функциями переходов и выходов, определяет свое следующее состояние, выходной символ \overline{b} , и перемещается на вектор \overline{b} .

В момент $(2\tau+1)$ $(\tau\in\mathbb{N}_0)$ хищник W_j воспринимает в качестве входного символа пару $(\mathcal{F}_1,\mathcal{F}_2)$, задающую расположения жертв в зоне обзора W_j \langle и расположения и состояния других хищников, находящихся в зоне обзора W_j \rangle в моменты 2τ и $(2\tau+1)$, и, в соответствии со своими функциями переходов и выходов, определяет свое следующее состояние, выходной символ \overline{b} , и перемещается на вектор \overline{b} .

Система хищников K "ловит" жертву, если жертва в некоторый момент времени оказалась в окрестности хода одного из хищников. Пойманная жертва исчезает с плоскости. K "ловит" независимую систему жертв, если в процессе преследования K ловит каждую жертву.

Ставится вопрос: существует ли независимая система \langle коллектив \rangle хищников K(R,V) и их начальное расположение на плоскости, такое что для произвольной конечной независимой системы жертв S(R',V') и любого начального расположения жертв K ловит S.

Расположение автоматов системы, при котором все они находятся в одной клетке, назовем κa ноническим. Зафиксируем $R, V \in \mathbb{N}$, такие что $2 \le V \le R$.

Теорема 1. При любых натуральных R', V' ($R' \geq V'$), для любой независимой системы хищников $K = (W_1, \ldots, W_m)(R, V)$, любого начального расположения W_1, \ldots, W_m и любого автоматажертвы U = U(R', V') существует начальное расположение U, при котором K не ловит U.

Теорема 2. Имеют место утверждения:

- 1) для любого автомата-жертвы U=U(R,V-1) существует коллектив хищников $K=(W_1,W_2,W_3)(R,V)$, который, стартуя из канонического расположения, ловит U при любом начальном взаимном расположении U и K;
- 2) при любых натуральных R', V', таких что $R' \geq V'$, для любого коллектива $(W_1, W_2)(R', V')$, любого автомата-жертвы U = U(R, V 1) и любого начального расположения $(W_1, W_2)(R', V')$, существует такое начальное расположение U, что (W_1, W_2) не ловит U.

Теорема 3. Существует коллектив хищников $K = (W_1, \ldots, W_{28})(R, V)$, который, стартуя из канонического расположения, ловит любую конечную независимую систему жертв $S = (U_1, \ldots, U_n)(R, V - 1)$ при любом начальном взаимном расположении U_1, \ldots, U_n и K.

- 1. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. "Введение в теорию автоматов", Наука, 1985.
- 2. Г. Килибарда, В. Б. Кудрявцев, Ш. Ушчумлич "Независимые системы автоматов в лабиринтах", Дискретная математика, т. 15 вып. 2, 2003.
- 3. Г. Килибарда, В. Б. Кудрявцев, Ш. Ушчумлич "Коллективы автоматов в лабиринтах", Дискретная математика, т. 15 вып. 3, 2003.
- 4. Грунская В. И., "О динамическом взаиомдействии автоматов", в кн.: Математическая кибернетика и ее приложения к биологии, МГУ, 1987, стр. 8-18.

О решении задачи слежения за динамическими объектами в условиях помех

Волченков М. П., volch max@mail.ru

кафедра МаТИС, механико-математический факультет МГУ им. Ломоносова

Задачи автоматического анализа изображений и сопровождения динамических объектов возникают во многих областях техники. В качестве динамических объектов могут рассматриваться летательные аппараты, облака, автомобили на шоссе, двигающееся лицо человека и т.п. Решение данных задач в общем виде комбинаторно сложно ввиду значительного числа рассматриваемых точек последовательных изображений. Мы рассматриваем только особые (характерные) точки, значимо отличающиеся от своих соседей, и отождествляем объект с множеством таких точек, а движения объекта устанавливаем методом нахождения соответствия между особыми точками разных кадров.

При выделении особых точек пороговыми методами, как правило, число и состав таких точек может меняться от кадра к кадру, что принципиально затрудняет задачу. Предлагаемая модель допускает случаи, когда число характерных точек на соседних кадрах может быть различно. При этом на последовательности кадров возможно как пропадание имеющихся характерных точек, так и появление новых.

Формально возникновение и пропадание точек реализуется следующим образом. Искусственно добавляются точки с неопределёнными координатами и минимальной яркостью. Далее для двух групп характерных точек (x_i, y_i, δ_i) , (v_i, w_i, Δ_i) , $i = \overline{1,n}$, где (x_i, y_i, δ_i) - абсцисса, ордината и яркость i -ой точки первого кадра, а (v_i, w_i, Δ_i) - абсцисса, ордината и яркость i -ой точки второго кадра, рассматривается функционал

$$\sum_{i=1}^{n} \left((x_i - v_{\sigma(i)})^2 + (y_i - w_{\sigma(i)})^2 + \alpha \cdot (\delta_i - \Delta_{\sigma(i)})^2 \right),\,$$

в котором искусственно добавленная точка $(v_{\sigma(i)}, w_{\sigma(i)})$ ассоциируется с точкой (x_i, y_i) .

Целью задачи является нахождение перестановки $\sigma:\{1,\ldots,n\}\to\{1,\ldots,n\}$ такой, что значение указанного выше функционала минимально. Здесь α - коэффициент относительной значимости яркости.

Предложен алгоритм, который позволяет свести задачи в рассмотренной модели к задачам поиска минимальных паросочетаний [6]. Сложность алгоритма составляет $O(n^3)$.

В результате точки первого кадра, сопоставленные добавленным точкам второго кадра, считаются пропавшими; точки второго кадра, сопоставленные добавленным точкам первого кадра, считаются появившимися. Такой подход позволяет обрывать и завязывать траектории характерных точек на протяжении всей последовательности кадров.

Алгоритм допускает модификацию сложности порядка $O(n^2)$ для получения приближённого решения [3], при этом оно отличается от точного не более чем в 2 раза. Алгоритм был опробован на реальных данных и показал хорошее качество работы в условиях помех.

Автор выражает благодарность академику Кудрявцеву В. Б. и проф. Бабину Д. Н. за ценные замечания и внимание к работе.

- [1] М. П. Волченков, И. Ю. Самоненко. "Об автоматическом распознавании лиц". // Интеллектуальные системы, в печати.
- [2] М. П. Волченков. "Опыт использования гауссовых смесей для выделения характерных точек и решения задачи слежения". // РФЯЦ ВНИИЭФ. Молодежь в науке. Саров 2005.
- [3] А. Е. Годин. "Обобщенная задача о назначениях". // Материалы международного семинара по супервычислениям и математическому моделированию. Саров 2002.
- [4] Э. Майника. "Алгоритмы оптимизации на сетях и графах". // Мир. Москва 1981.
- [5] Л. Р. Форд, Д. Р. Фалкерсон. "Потоки в сетях". // Мир. Москва 1966.
- [6] H. W. Kuhn. The Hungarian Method of the Assignment Problem. // Naval Research Logistics Quart, 1955.

Автоматные модели защищенных компьютерных систем

Галатенко А.В.,

кафедра MaTUC механико-математического факультета, $M\Gamma Y$ им. M.~B.~Jомоносова

В работе рассматривается автоматная система, часть состояний которой объявляется безопасной. Исследуются языки, не выводящие автомат из класса безопасных состояний или выводящие из класса безопасных состояний не более, чем на ε .

Под конечным автоматом мы будем понимать четверку $V=(A,Q,\varphi,q_0)$, где A — конечное множество входных символов, Q — конечное множество состояний, $\varphi:A\times Q\to Q$ — функция переходов, $q_0\in Q$ — начальное состояние. Пусть $Q=S\cup I$, причем $S\cap I=\emptyset$. Состояния из S назовем безопасными, состояния из I — небезопасными. Далее будем предполагать, что начальное состояние является безопасным.

Обозначим через A^* множество всех конечных слов в алфавите A. Функция φ может быть продолжена на множество $A^* \times Q$ по мультипликативности. Подмножество A^* называется языком. Каждому слову $\alpha \in A^*$ соответствует слово $\kappa(\alpha) \in Q^*$, $\kappa(\alpha) = \varphi(\alpha, q_0)$. Назовем слово $\alpha \in A^*$ безопасным, если $\kappa(\alpha) \in S^*$. Назовем язык $A \subseteq A^*$ безопасным (S-языком), если все слова, составляющие A, безопасны, и не существует безопасных слов, не принадлежащих A.

Сформулируем ряд утверждений о свойствах S-языков.

Лемма 1. Если A является S-языком, то A регулярен.

Пусть \mathcal{A} — некоторый язык. Обозначим через $[\mathcal{A}]$ множество всех непустых начал слов, принадлежащих \mathcal{A} .

Лемма 2. Язык \mathcal{A} является S-языком для некоторого автомата V тогда и только тогда, когда выполнены следующие условия:

- 1. *А* непуст;
- 2. А регулярен;
- 3. $[A] \subseteq A$.

Лемма 3. Автоматно перечислимые языки (см. [3]) являются S-языками, но, вообще говоря, не наоборот.

Содержательно безопасный язык может интерпретироваться несколькими способами:

- в системах активного аудита (см. например [1]), как описания легального поведения пользователей (в этом случае выход из множества безопасных состояний трактуется как атака);
- при задании моделей злоумышленника (в этом случае выход из множества безопасных состояний может трактоваться как обнаружение злоумышленника);
- при построении моделей гарантированно защищенных систем (см. например [4], [2]; в таких моделях доказательства как правило проводятся индуктивно, а лемма 2 означает, что S-языки как раз описывают все регулярные языки, для которых проходит индуктивное доказательство).

Оценим число S-языков, генерируемых одним автоматом. Будем предполагать, что все состояния автомата достижимы из q_0 . Обозначим через NS_V число S-языков, которые можно задать в автомате V.

Лемма 4. Для конечного автомата V с n состояниями имеет место $NS_V \ge n$, u эта оценка неулучшаема.

Определим функцию Шеннона LS(n) числа S-языков, задаваемых автоматами с n состояниями, следующим образом. $LS(n) = \max_{\{V:|O|=n\}} (NS_V)$.

Теорема 1. Для любого $\varepsilon > 0$ существуют константы C_1, C_2, D и N такие, что

$$C_1 \times D^{|A| \times log_{|A|}(n)} \le LS(n) \le C_2 \times (D \times (1+\varepsilon))^{|A| \times log_{|A|}(n)}.$$

для всех $n \ge N$.

Рассмотрим произвольное $\varepsilon > 0$. Введем функции $s: Q^* \to \mathbb{N} \cup \{0\}$ и $i: Q^* \to \mathbb{N} \cup \{0\}$ следующим образом. Пусть $\kappa \in Q^*$. $s(\kappa)$ равняется числу букв κ , содержащихся в S, $i(\kappa)$ равняется числу букв κ , содержащихся в S, $i(\kappa)$ равняется числу букв κ , содержащихся в S, $i(\kappa)$ равняется числу букв K, содержащихся в S, $i(\kappa)$ равняется числу K, содержащих K, содержащих

 S_{ε} -языки являются естественным обобщением S-языков. Содержательно рассмотрение возможности выхода из множества безопасных состояний может интерпретироваться как признание ущерба, не превышающего некоторого порога, допустимым. Отметим, что свойства S_{ε} -языков существенно отличаются от свойств S-языков.

Теорема 2. Если $\varepsilon \in \mathbb{Q}$, $0 < \varepsilon < 1$, то существует конечный автомат V такой, что определяемый им S_{ε} -язык является контекстно-свободным, но не регулярным, и существуют конечные языки, не являющиеся S_{ε} -языками ни при каких $\varepsilon \in \mathbb{Q}$.

Замечание. При $\varepsilon = 0$ мы остаемся в множестве S-языков. $S_1 = A^*$.

Теорема 3. Если $\varepsilon_1, \varepsilon_2 \in \mathbb{Q}, \ 0 \le \varepsilon_1 < \varepsilon_2 \le 1$, то существует язык, являющийся S_{ε_1} -языком, но не являющийся S_{ε_2} -языком.

Автор выражает глубокую благодарность своему научному руководителю, д.ф.-м.н., проф. В.Б. Кудрявцеву за постановки задач и внимание к работе.

- 1. Галатенко А.В., "Активный аудит". JetInfo №8, 1999, http://www.jetinfo.ru/1999/8/1/article1.8.1999.html
- 2. Галатенко А.В., "Об автоматной модели защищенных компьютерных систем". "Интеллектуальные системы", т. 4, вып. 3–4, Москва, 1999г., с. 263–270.
- 3. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. "Введение в теорию автоматов— М.: Наука, 1985.
- 4. Moskowitz I.S., Costich O.L., "A Classical Automata Approach to Noninterference Type Problems". Department of the Navy, Naval Research Laboratory, 1992.

Разработка когнитивной модели пользователя

Гарифуллина Ю., тел. (812) 336-8970, E-mail: <u>juli@big.spb.ru</u>
Санкт-Петербургский Государственный Политехнический Университет
Факультет Управления и Информационных Технологий
Кафедра Компьютерных Интеллектуальных Технологий в Проектировании
195251, Санкт-Петербург, ул. Обручевых, 1,

В статье описан подход к разработке когнитивной модели пользователя с использованием возможностей онтологического инжиниринга. Рассмотрен один из когнитивных процессов – внимание.

Ключевые слова: User modeling, cognitive, ontology.

Онтология пользователя

С точки зрения онтологического инжиниринга [Гаврилова, Хорошевский, 2001] определим два подхода к построению модели пользователя:

- 1. От общего к частному: создается глобальная онтология понятия пользователь. Такая онтология содержит бесконечное число узлов, и, следовательно, описана полностью быть не может никогда. Однако на определенном уровне зрелости такая онтология может породить множество конечных онтологий, например, когнитивную, социальную, онтологию знаний. Пример реализации такого подхода представлен на [www.gumo.org].
- 2. От частного к общему: создаются частные конечные онтологии, например, те же когнитивная онтология, социальная, онтология знаний. Они объединяются в одну глобальную онтологию. Добавление каждой новой частной наращивает глобальную онтологию.
- В условиях разработки портала дистанционного обучения, особый интерес представляет когнитивная модель пользователя.

Когнитивная психология

Когнитивная психология изучает то, как люди получают информацию о мире, как эта информация представляется человеком, как она хранится в памяти и преобразуется в знания и как эти знания влияют на наше внимание и поведение [Солсо, 2002].

Когнитивная психология охватывает широкий диапазон психологических процессов – от ощущений до восприятия, распознавания образов, внимания, обучения, памяти, формирования понятий, мышления, воображения, запоминания, языка, эмоций и процессов развития.



Рис.2. Основные направления исследований в когнитивной психологии [Солсо, 2002]

Внимание

В когнитивной психологии внимание это концентрация умственного усилия на сенсорных или мысленных событиях [Солсо, 2002]. Это процесс упорядочивания поступающей извне информации в контексте стоящих перед пользователем задач [Шнейдерман, 1984]. В исследованиях внимания рассматриваются четыре главных аспекта: пропускная способность и избирательность внимания, уровень возбуждения, управление вниманием и сознание. Для каждого аспекта определяется характеристика пользователя, способ получения информации о пользователе, способ представления информации и параметры адаптивного интерфейса, влияние на которые оказывает характеристика. Поэтому когнитивная онтология пользователя содержит помимо связи «ребенок-родитель» специальные типы связей: метод получения информации, модель представления, характеристика, параметр адаптивного интерфейса. Способность человека к обработке информации ограничена "пропускной способностью канала", поэтому внимание действует избирательно. Такое представление предполагает, что где-то в переработке информации есть "узкое место", частично обусловленное неврологическими

-

¹ Работа поддержана грантом Бел-РФФИ N 06-01-81005

возможностями. Предложено несколько моделей, в которых определены функция и локализация этого узкого места: модель делителя, модель с фильтрацией, модель уместности [Солсо, 2002]. Чем ниже пропускная способность канала (уже "узкое место"), тем меньше сенсорных признаков одновременно может обрабатывать человек. В адаптивном интерфейсе сенсорными признаками являются возможности мультимедиа, текст, доступные ссылки, опции навигации, элементы помощи [Брусиловский, 1996]. Для пользователей с низким уровнем пропускной способности внимания доступные на данный момент возможности мультимедиа должны ограничиваться самыми необходимыми

Для определения избирательности внимания существует "методика Мюнстерберга" [Лучшие психологические тесты, 1992].



Рисунок 4. Частная онтология "Пропускная способность и избирательность"

Возбуждение поддерживает нашу способность к восприятию сенсорных сигналов в активном состоянии. Повышение уровня возбуждения сопровождается ухудшением деятельности вплоть до определенной точки, после которой она вновь ухудшается. График этой зависимости имеет вид перевернутой буквы "U" и называется законом Йеркса-Додсона. [Солсо, 2002] Связь между деятельностью и возбуждением обусловлена тем, как используются те или иные сенсорные признаки. Существует предположение, что по мере роста возбуждения, диапазон сенсорных признаков, используемых для управления действием, ограничивается. При низком возбуждении избирательность тоже низка: внимание привлекается как существенными, так и несущественными признаками. С ростом возбуждения происходит сосредоточение на существенных признаках, а несущественные игнорируются. Однако при дальнейшем росте возбуждения избирательность становится столь узкой, что игнорируются даже некоторые существенные признаки. Это та точка, где деятельность перестает улучшаться и ее показатели снижаются. Возбуждение является постоянно меняющейся характеристикой и может учитываться только в динамически модифицируемых моделях.

Оценить объем воспринимаемой пользователем информации позволяет методика "корректурная проба" [Лучшие психологические тесты, 1992].



Рисунок 5. Частная онтология "Уровень возбуждения и/или интерес"

Управление рассматривается как элемент модели памяти, предложенной Аткинсоном и Шифриным [Солсо, 2002]. В их модели память – большая и постоянная совокупность узлов. А автоматическая обработка сенсорных признаков начинается, когда предъявляется стимул (например, буква). Человек в какой-то степени контролирует этот процесс. Он может направить свое внимание на какую-то информацию или отвлечься от нее. Пользователь, способный в лучшей мере управлять процессами обработки стимулов, может работать с большим количеством сенсорных признаков. В зависимости от уровня контроля пользователя над процессами внимания, ему может быть предоставлено больше или меньше возможностей, более или менее сложные текстовые блоки, механизм сортировки или механизм сокрытия ссылок, более подробная или более тщательно прореженная карта навигации и т.д.

Оценить способность пользователя к управлению процессами внимания (или переключать внимание) можно при помощи методики "красно-черная таблица" [Лучшие психологические тесты, 1992].



Рисунок 5. Частная онтология "Уровень возбуждения и/или интерес"

Сознание в контексте внимания характеризует знание о событиях или стимулах окружающей среды, а так же знание о когнитивных явлениях, таких как память, мышление и телесные ощущения [Солсо, 2002]. У человека могут работать два типа сознания: одно — для речевого знания и обработки информации в левом полушарии, а другое — для пространственных функций, локализованных в правом полушарии. Более слабое полушарие, у которого есть свои уникальные возможности, не полностью выполняет свои функции, поскольку оно забито более сильным. В моменты наивысшего напряжения, доминантное полушарие принимает управление. Эксперименты показывают, что с функционированием полушарий тесно связаны процессы внимания, при этом левое полушарие обеспечивает селективное (избирательное) внимание, а правое — поддержку общего уровня настороженности [Шнейдерман, 1984].

Тема сознания не поддается легким аналитическим измерениям, поэтому ощущается острый недостаток теоретических моделей в этой области. Исключение составляют работы Эндела Тульвинга, который предложил выделить три разновидности сознания: аноэтичное, ноэтичное и автоноэтичное. Каждой из них соответствуют три вида памяти: эпизодическая, семантическая и процедурная. Взаимоотношения между системами памяти и типами сознания приведены в [Солсо, 2002].



Рисунок 6. Частная онтология "Сознание"

Полная онтология представляет собой взаимосвязь аспектов понятия внимание, методов получения информации по этим аспектам, а так же моделей, или форм представления этой информации. Аналогичное построение онтологий других областей когнитивной психологии позволит создать полную когнитивную модель пользователя.

Реализовать построение модели пользователя, а так же собрать информацию для модели при помощи тестирования позволяет инструмент моделирования пользователя Trivium, разработанный студентами кафедры Компьютерных Интеллектуальных технологий в Проектировании Санкт-Петербургского Государственного Технического университета под руководством проф. д.т.н. Гавриловой Т.А. Инструмент моделирования Trivium позволяет строить и заполнять модели пользователя произвольной структуры, а так же поддерживает стереотипный подход [Брусиловский, 1996]. Разработка когнитивной модели с использованием Trivium позволит создать тиражируемую когнитивную модель пользователя с необходимым для ее заполнения набором тестов и методик.

- 1. Лучшие психологические тесты для профотбора и профориентации, ответственный редактор Кудряшов А. Ф., Петрозаводск, издательство "Петроком", -319 с.
- 2. Солсо Р., Когнитивная психология, Санкт-Петербург, издательство "Питер", 591 с.
- 3.Шнейдерман Б., Психология программирования. Человеческие факторы в вычислительных и информационных системах, Москва, издательство "Радио и связь", 303 с
- 4. Брусиловский П. Технологии и методы адаптивной гипермедиа, User Modeling and User Adapted Interaction, v 6, n 2-3, стр. 87-129
- 5. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем, Санк-Петербург, издательство "Питер", -384 с.

Точное значение сложности угадывания одного множества сверхслов

Э. Э.Гасанов, А. В.Лавриненко

Понятие угадывающего автомата впервые было введено в [1].

Пусть $\{0,1\}^*$ и $\{0,1\}^\infty$ — соответственно множества слов и сверхслов в алфавите $\{0,1\}$. Пусть A — конечный автомат, преобразующий входное сверхслово $a=a_1a_2\dots a_t\dots\in\{0,1\}^\infty$ в выходное сверхслово $b=b_1b_2\dots b_t\dots\in\{0,1\}^\infty$. Скажем, что автомат A угадывает сверхслово $a=a_1a_2\dots a_t\dots$, если существует такой номер N, что для любого номера $n\geq N$ выполнено $b_n=a_{n+1}$.

Пусть B некоторое множество сверхслов. Скажем, что автомат yгаdывает mножество сверхслов B, если он угадывает каждое сверхслово из множества B.

Скажем, что множество сверхслов угадываемо, если существует угадывающий его автомат.

Если B — угадываемое множество сверхслов, то через ||B|| обозначим число состояний у минимального по числу состояний автомата, угадывающего множество B, и назовем число ||B|| сложностью угадывания множества сверхслов B.

Слово a называется henepuoduческим, если не существует такого слова <math>b, что $a=b^n$, где n>1. Обозначим через R_t — множество всех непериодических слов длины t в алфавите $\{0,1\}$. Известно, что мощность R_t определяется формулой

$$|R_t| = \sum_{d|t} \mu(d) 2^{rac{t}{d}},$$
где $\mu(x)$ – функция Мебиуса.

Обозначим через \widehat{R}_t — множество всех сверхслов с произвольными предпериодами и периодами из R_t , т.е. $\widehat{R}_t = \{ab^{\infty} : a \in \{0,1\}^*, b \in R_t\}$.

В [2] показано, что угадываемы только множества периодических сверхслов с ограниченным периодом, и для таких множеств сверхслов получен порядок сложности угадывания.

В частности, из нижней оценки, полученной в [2], следует, что $||R_t|| \ge |R_t|/2$.

Авторам данной работы удалось построить автомат, угадывающий множество \widehat{R}_t , сложность которого совпадает с нижней оценкой для нечетных t, и отличается от нее на 1 при четных t. Тем самым имеет место следующая теорема.

Теорема 1 Для нечетного t имеет место $||\widehat{R}_t|| = |R_t|/2$, a для четного t выполнено $|R_t|/2 \le ||\widehat{R}_t|| \le |R_t|/2 + 1$.

Список литературы

- 1. Вереникин А. Г., Гасанов Э. Э. Некоторые оценки сложности угадывающего автомата. *Тезисы* докладов XIV Международной конференции "Проблемы теоретической кибернетики" (Пенза, май 2005).
- 2. Вереникин А. Г., Гасанов Э. Э. Об автоматной детерминизации множеств сверхслов. Дискретная математика (2006) 18, № 2.

О синтезе синхронизирующих деревьев

Гасанов Э. Э., Проворова А. Л.

В данной работе рассматривается известная проблема синхронизации сигнала, возникающая при производстве электронных чипов. Она состоит в том, чтобы от некоторой точки чипа (источника) до некоторых других точек чипа сигнал доходил одновременно. Эта задача решалась как для чисто производственных алгоритмов [1], так и на модельных объектах[2,3,4]. Модель, рассматриваемая в данной работе отличается от моделей из [2,3,4] тем, что она более приближена к реальным физическим процессам.

Мы будем рассматривать ориентированные деревья с корнем, каждая вершина которых лежит на плоской целочисленной решетке, каждое ребро соединяет две соседние вершины целочисленной решетки (т.е. каждое ребро имеет длину 1, а степень инцидентности каждой вершины не более 4), и все ребра направлены от корня к концевым вершинам, при этом корень — это вершина, в которую не входит ни одно ребро, а из концевых вершин не исходит ни одного ребра. На рисунке 1 приведен пример такого дерева, причем корень дерева помечен крестиком, и у него имеется 3 концевые вершины. Ориентация ребер на рисунке не приведена, так как после фиксации корня она определяется однозначно.

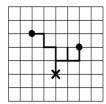


Рис. 1: Пример синхронизирующего дерева.

Полустепенью исхода вершины дерева назовем число ребер, исходящих из вершины. Задержкой вершины дерева назовем ее полустепень исхода, задержкой пути в дереве — сумму задержек всех вершин пути, а задержкой до концевой вершины дерева — задержку пути, ведущего от корня к этой концевой вершине.

Почувствовать понятия задержек вершины и пути проще всего в "водной" интерпретации. В этой интерпретации каждая вершина представляет собой водный насос, ребра — это водные каналы (арыки). Для каждой вершины (кроме корня) один канал (входящее ребро) является управляющим, и если он полностью заполняется водой, то насос вершины включается и начинает заполнять (из своего резервуара) выходные каналы (соответствующие исходящим ребрам, которых может быть не более 3), причем все выходные каналы вершины заполняются одновременно. Время заполнения выходных каналов можно считать равным их числу (полустепени исхода вершины) и называть задержкой данной вершины. Соответственно задержка пути до концевой вершины — это время, через которое заполнится входной (управляющий) канал этой вершины после того как включится насос корневой вершины. Оно равно суммарной задержке пути и есть время доставки сигнала от корня до этой концевой вершины.

Так введенная задержка пути и есть главное отличие данной модели от моделей из [2,3,4], в которых задержка пути просто равна его длине.

Конфигурацией назовем множество точек плоской целочисленной решетки, одна из которых называется источником, а остальные точки конфигурации называются стоками. На рисунке 1 приведен пример конфигурации из трех точек, причем источник помечен крестиком, а стоки жирными точками.

Основная задача: для заданной конфигурации точек K построить дерево A, корень которого совпадает с источником, множество концевых вершин которого содержит множество стоков конфигурации К и задержка до всех концевых вершин дерева, являющихся стоками конфигурации, одинакова. Дерево A, построенное для конфигурации K и обладающее данными свойствами, будем называть *синхронизирующим* и обозначать A(K).

Неформально, синхронизирующее дерево позволяет доставлять сигнал от источника за одинаковое время до каждого из стоков. На рисунке 1 приведен пример синхронизирующего дерева, для определенной ранее конфигурации.

Отметим, что в синхронизирующем дереве могут быть концевые вершины не совпадающие со стоками. Наличие таких вершин является второй отличительной чертой предлагаемой модели и служит дополнительным средством, помогающим регулировать время доставки сигнала.

Paccmoянием между двумя точками плоскости $a=(x_a,y_a)$ и $a'=(x_{a'},y_{a'})$ назовем число

 $ho(a,a') = |x_a - x_{a'}| + |y_a - y_{a'}|.$ $r(K) = \min_{a,a' \in K, \ a \neq a'}
ho(a,a')$ — минимальное расстояние между любыми двумя разными точками конфигурации K.

Для натурального т введем следующий класс конфигураций

$$\mathcal{K}_m = \{K : r(K) \ge m\}.$$

Конфигурацию точек, для которой невозможно построить синхронизирующее дерево, будем называть ловушкой.

Теорема 2 В классах \mathcal{K}_1 и \mathcal{K}_2 существуют ловушки.

На рисунке 2 приведены ловушки для классов \mathcal{K}_1 и \mathcal{K}_2 .

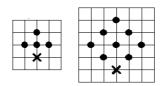


Рис. 2: Ловушки для \mathcal{K}_1 и \mathcal{K}_2 .

Теорема 3 Для любой конфигурации из K_n , где $n \ge 5$, существует синхронизирующее дерево.

Идея алгоритма построения синхронизирующего дерева для конфигурации из \mathcal{K}_n , где $n \geq 5$, состоит в следующем. Сначала вводится понятие регулярной 5-конфигурации, у которой все соседние точки находятся на расстоянии 5. На рисунке 3 приводится фрагмент регулярной 5-конфигурации.

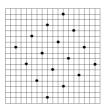


Рис. 3: Регулярная 5-конфигурация.

Далее предлагается способ построения синхронизирующего дерева для регулярной 5-конфигурации. На рисунке 4 приведено синхронизирующее дерево для 8 точек регулярной 5-конфигурации, а на рисунке 5 схематически изображено синхронизирующее дерево для регулярной 5-конфигурации.

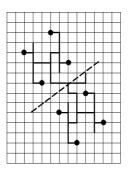


Рис. 4: Синхронизирующее дерево для 8 точек регулярной 5-конфигурации.

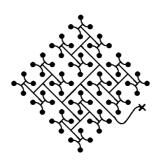


Рис. 5: Синхронизирующее дерево для регулярной 5-конфигурации.

Далее берется регулярная 5-конфигурация покрывающая всю плоскость. Плоскость разбивается на одинаковые по форме элементарные области, как показано на рисунке 6, так что к каждую область попадает только одна точка конфигурации, причем эта точка является центром области.

Далее рассматривается произвольная конфигурация $K \in \mathcal{K}_5$. По определению в каждую элементарную область попадает не более одной точки из конфигурации K. Центры областей, в которые попадают точки конфигурации K, образуют вспомогательную конфигурацию K', являющуюся фрагментом регулярной 5-конфигурации. Для конфигурации K' по описанному ранее методу строится

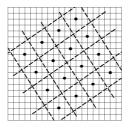


Рис. 6: Разбиение плоскости на элементарные области.

синхронизирующее дерево, которое будет планом синхронизирующего дерева для конфигурации K. Далее этот план корректируется, чтобы доставить сигнал до точек K и обеспечить одновременность этого сигнала. Перебором всех возможных вариантов показывается, что такая коррекция всегда возможна.

На данный момент остается открытым вопрос о наличии ловушек в классах \mathcal{K}_3 и \mathcal{K}_4 .

Список литературы

- 1. Pavisic I., Lu A., Zolotykh A.A., Gasanov E.E. Method in integrating clock tree synthesis and timing optimization for an integrated circuit design. United States Patent: 6,550,044, April 15, 2003
- 2. N.A.Sherwani. Algorithms for VLSI Physical Design Avtomation. Kluwer Academic Publishers, 1993.
- 3. L.A.Fisher, H.T.Kung. Synchronizing Large Systolic Arrays. Proceedings of SPIE, pp. 44-52, May 1982.
- 4. A.Kahng, J.Cong, G.Robins. High-Performance Clock Routing Based on Recurcive Geometric Matching. Proceedings of 28th ACM/IEEE Design Automation Conference, pp. 322-327, June 1991.

Об автоматной модели самоочищения легких

Ю. Г. Гераськина

Универсальность автоматного моделирования широко используется прежде всего в технике, однако в последнее десятилетие его все шире начинают использовать в биологии. Примеры такого рода доставляют попытки автоматного описания работы сердца, нейронного управления организма, а также клетки.

В предлагаемой работе рассматривается возможность представления легочного механизма живых систем с помощью автоматов.

Легкие образуют древовидную структуру бронхов, в которых имеются реснички, играющие роль эскалаторного механизма вывода как внутреннего секрета, так и поступающего извне в легкие вещества, во внешнюю среду. Бронхи имеют разные пропускные способности и разную эффективность ресничек. Чем выше от альвеол, то есть самых мелких бронхов, тем мощнее механизм передачи вещества изнутри вовне.

Учитывая выше изложенное, будем представлять легкие полным дихотомическим ориентированным к корню деревом, которое будем называть I-depeвом и обозначать D^{-1} , со следующими параметрами.

Пусть $\mathbb N$ - множество натуральных чисел и $l, l \in \mathbb N$, считаем глубиной этого І-дерева. Полагаем, что ребро І-дерева D^{-1} , инцидентное корню, имеет глубину 1.

Каждое ребро из D^{-1} разделено на $n, n \in \mathbb{N}$, равных частей, называемых *ресничками*, и занумерованных числами $i, i = 1, 2, \ldots, n$, возрастающими в направлении, обратном ориентации ребра.

Припишем каждому ребру глубины $j, j = 1, 2, \ldots, l$, два числа $2^{l-j}b$ и $2^{l-j}r$, где $b, r \in \mathbb{N}$ и $r \leq b$, называемых максимальной нагрузкой и мерой переброса ресничек ребер глубины j соответственно. Такое I-дерево D^{-1} с описанными выше параметрами b, r, n и l обозначим $D^{-1}(b, r, n, l)$. Свяжем с ним некоторый процесс, который назовем процессом дыхания. Он обусловлен рядом допущений.

Считаем, что в $D^{-1}(b,r,n,l)$ заданы распределения значений нагрузок по всем ресничкам, учитывая, что нагрузка может быть нулевой. Пусть V' — суммарная нагрузка по всем ресничкам, а V — максимально возможная суммарная нагрузка по всем ресничкам. V назовем объемом I-дерева (легких), а V' — исходным объемом загруженности I-дерева.

I-дерево $D^{-1}(b, r, n, l)$ с исходным объемом загруженности V' обозначим $D^{-1}(b, r, n, l; V')$.

Пусть $\delta \in (0;1]$, тогда *предельно допустимым порогом* будем называть объем загруженности І-дерева $D^{-1}(b,r,n,l;V')$, равный $]\delta V[$, то есть наименьшему натуральному числу, не меньшему, чем δV . Далее будем писать δV вместо $]\delta V[$, считая, что $\delta V \in \mathbb{N}$.

Каждая ресничка осуществляет прием вещества извне и переброс своей нагрузки на следующую ресничку с меньшим номером внутри ребра.

Прием ресничкой вещества, имеющего массу $d, d \in \mathbb{N}_0$ и $d \leqslant V - V'$, из внешней среды внутри ребра осуществляется по следующему правилу (для этого правила ориентация считается обратной к заданной).

- А₁) Если ресничка имеет максимальную нагрузку, то прием вещества не осуществляется.
- $\mathsf{G}_1)$ При немаксимальной нагрузке d_1 первой такой реснички она осуществляет прием вещества максимально возможной массы d_2 , такой, что $d_1+d_2\leqslant \min(b,d)$, где b максимальная нагрузка этой реснички.
 - B_1) Следующая за ресничкой из B_1) принимает массу d_3 , как и в B_1), с заменой там d на $d-d_2$.
- Γ_1) Оставшаяся масса вещества опускается до следующей реснички с большим номером в ребре, для которой не выполняется условие A_1). Она осуществляет прием вещества по правилу B_1) или B_1).
- \mathcal{A}_1) Если ресничка в рассматриваемом ребре является последней, не удовлетворяющей условию A_1), то оставшаяся масса вещества делится пополам (если число нечетное, то одна из частей на единицу больше другой); и каждая из частей вещества воспринимается соответствующими ребрами, как описано выше.
- E_1) Процесс, описываемый позициями A_1)— J_1), начинается с ребра, которое инцидентно корню. Переброс ресничкой вещества осуществляется на следующую ресничку с меньшим номером внутри ребра по такому правилу.
- A_2) Если следующая ресничка имеет не нулевую нагрузку, то переброс с реснички не осуществляется.
- G_2) Если нагрузка реснички не превосходит r, где r ее мера переброса, и не выполнено условие A_2), то перебрасывается на следующую вся нагрузка реснички и считается, что ее нагрузка становится равной нулю.
- B_2) Если на ресничке нагрузка m и m > r, то она перебрасывает на следующую ресничку нагрузку r и оставляет у себя нагрузку m r.

Если ресничка в ребре последняя, то переброс нагрузки осуществляется по правилам A_2), B_2), B_2).

- Γ_2) Если ребро инцидентно корню, то переброс с наименьшей по номеру реснички осуществляется в среду по правилам B_2) и B_2) в предположении, что среда играет роль реснички с нулевой нагрузкой.
- \mathcal{A}_2) Если ребро не инцидентно корню, то есть его вершина инцидентна следующему ребру, то нагрузка с наименьшей по номеру реснички этого ребра передается наибольшей по номеру ресничке другого ребра по правилам A_2), B_2).

Считаем, что процесс дыхания осуществляется в дискретные моменты времени $t=1,2,3,\ldots$

В первый момент I-дерево $D^{-1}(b,r,n,l;V')$ имеет заданное распределение нагрузок по его ресничкам.

Ко второму моменту осуществляется прием вещества массой d(1) по правилам A_1)– E_1), и затем осуществляется переброс нагрузок с реснички на ресничку во всем I-дереве или выброс в среду в соответствии с правилами A_2), B_2), B_2), B_2), B_2), B_2). А если в легкие подается масса d, не превосходящая объема легких, то та ее часть, которая не осела на ресничках, выбрасывается в среду. Другими словами, за один момент (шаг) происходит «вдох» и «выдох».

Если в каждый момент времени $t=1,2,3,\ldots$ все реснички I-дерева $D^{-1}(b,r,n,l;V')$ осуществляют прием вещества нулевой массы, то такой процесс называется *процессом самоочищения* этого I-дерева. Процесс самоочищения заканчивается в такой момент времени t, в котором нагрузки всех ресничек I-дерева $D^{-1}(b,r,n,l;V')$ впервые стали равными нулю.

Под распределением загрузки V' І-дерева $D^{-1}(b,r,n,l;V')$ будем понимать любое из возможных распределений нагрузок всех его ресничек таких, что суммарный объем их нагрузок равен V'. Ясно, что $V' \leq V$, где V - объем І-дерева $D^{-1}(b,r,n,l;V')$ и $V = 2^{l-1}bnl$. Такие распределения будем называть конфигурациями загрузки V'.

Пусть Q - множество всевозможных конфигураций загрузок $V',\ V' \leq V;\ Q_{\delta}$ - подмножество всех таких конфигураций из Q, загрузки V' которых не превосходят предельно допустимого порога δV , где $0<\delta<1$.

Введем функцию L(b,r,n,l,V') для І-дерева $D^{-1}(b,r,n,l;V')$, которая равна наибольшему из времен, за которое заканчивается процесс самоочищения $D^{-1}(b,r,n,l;V')$ при произвольном на-

чальном распределении загрузки V' этого І-дерева. Эту функцию обычно называют сложностной функцией Шеннона.

Теорема 1 Если І-дерево $D^{-1}(b,r,n,l;V')$ имеет объем V, равный $2^{l-1}bnl$, где $b,r,n,l,V,V' \in \mathbb{N}$ и $0 < V' \le V$, то функция L(b,r,n,l,V') в зависимости от значений параметров b,r,n,l и V' принимает следующие значения:

- a) $ecnu(2^l-1)bn \leq V' \leq V$, $mo L(b,r,n,l,V') = \frac{b}{r}[(2nl-1);$
- б) если $0 < V' < (2^l 1)bn$, то

 $npu \ r = 1 \ uмеем$

если $V' \leq bn, mo$

$$L(b,1,n,l,V') = \begin{cases} V' + b(n-1), & ecnu\ l = 1\ u\ n =]\frac{V'}{b}[, \\ 2V' -]\frac{V'}{b}[+ nl - 1, & unaue, \end{cases}$$

если $bn < V' \le bn + (l-1)n$, то L(b,1,n,l,V') = V' + n(l+b-1) - 1, если V' > bn + (l-1)n, то

$$L(b,r,n,l,V') = \begin{cases} \frac{b_{h_3}}{2^{l-1}} [+2b(nl-1), & \textit{ecau } k_3 = 1 \ \textit{u} \ h_3 = 1, \\ 2 \left(\frac{b_{h_3}}{2^{l-k_3}} [+(b-1)(n(l-k_3+1)-h_3)+nl-\frac{3}{2} \right), \ \textit{uhave}, \end{cases}$$

 $npu \ r > 1 \ uмеем$

если $V' \leq nl$, то L(b,r,n,l,V') = V' + nl - 1, если V' > nl, то

$$L(b,r,n,l,V') = \begin{cases} \frac{b_{h_3}}{2^{l-1}r} [+2] \frac{b}{r} [(nl-1), & \textit{ecau } k_3 = 1 \ \textit{u} \ h_3 = 1, \\ 2 \left(\frac{b_{h_3}}{2^{l-k_3}r} [+(] \frac{b}{r} [-1) (n(l-k_3+1)-h_3) + nl - \frac{3}{2} \right), \ \textit{uhave}, \end{cases}$$

где

$$k_3 = 1 + [l - log_2(\frac{V' - nl}{(b - 1)n} + 1)], h_3 = n -]\frac{V' - nl - (2^{l - k_3} - 1)(b - 1)n}{2^{l - k_3}(b - 1)}[+1, b_{h_3} = V' - nl - (2^{l - k_3} - 1)(b - 1)n - 2^{l - k_3}(b - 1)(n - h_3) + 1.$$

Пусть A^m — множество всех слов $\alpha^m = a(1)a(2)\dots a(m)$, где $a(t) \in \mathbb{N}_V \cup \{0\}$ при $t = 1, 2, \dots m$, на котором вводим отношение частичного порядка \leqslant , полагая $\alpha^m \leqslant \alpha'^m$, если $a(t) \leqslant a'(t)$ для всех $t = 1, 2, \dots, m$.

Для I-дерева $D^{-1}(b,r,n,l;V')$ слово α^m из A^m назовем donycmuмым, если при подаче на него буквы a(t) из α^m в каждый момент времени t всегда выполнено $a(t)+V(t)\leqslant \delta V$, где V(t) — объем загруженности I-дерева $D^{-1}(b,r,n,l;V')$ в момент $t,t=1,2,\ldots,m$. Такое слово называется npedenbho donycmumым для I-дерева $D^{-1}(b,r,n,l;V')$, если не существует допустимого слова α'^m из A^m такого, что $\alpha'^m \geqslant \alpha^m$ и $\alpha'^m \neq \alpha^m$.

Пусть $A^* = \bigcup_{m=1}^{\infty} A^m$, а A^{ω} — множество всех сверхслов (то есть слов бесконечной длины) $\alpha^{\omega} = a(1)a(2)\dots a(t)\dots$, где $a(t)\in\mathbb{N}_0$, на котором вводим отношение частичного порядка \leqslant , полагая $\alpha^{\omega}\leqslant\alpha'^{\omega}$, если $a(t)\leqslant a'(t)$ для всех t из \mathbb{N} .

Для I-дерева $D^{-1}(b,r,n,l;V')$ сверхслово α^{ω} из A^{ω} назовем donycmumum, если при подаче на него буквы a(t) из α^{ω} в каждый момент времени t всегда выполнено $a(t)+V(t)\leqslant \delta \mathbf{V}$, где V(t) — объем загруженности I-дерева $D^{-1}(b,r,n,l;V')$ в момент $t,t=1,2,\ldots$ Такое сверхслово называется donycmumum для double double дерева double double из double такого, что double из double из double дерева double из double такого, что double из double из double называется double на double по double на double на double на double по double на double на double на double по double на double на double на double на double на double по double на do

Ясно, что предельно допустимые сверхслова существуют. Например, сверхслово $\delta V - V' \delta V \delta V \dots \delta V \dots$ при $\delta V \leqslant r$ является предельно допустимым. Более того, можно показать, что для всякого допустимого сверхслова α^{ω} найдется предельно допустимое α'^{ω} такое, что $\alpha^{\omega} \leqslant \alpha'^{\omega}$.

Множество $\overline{A} = A^* \cup A^{\omega}$ будем называть множеством квазислов. Распространим понятия допустимости и предельной допустимости на квазислова.

Нашей задачей будет описание всех допустимых квазислов, а также выяснение того, будут ли множества допустимых и предельно допустимых слов и сверхслов регулярны и общегерулярны, соответственно, в автоматном смысле [3]. Ясно, что, указав все предельно допустимые квазислова, мы тем самым опишем все множество допустимых квазислов.

Занумеруем все реснички І-дерева $D^{-1}(b, r, n, l; V')$ таким образом, что ресничка с номером ijkявляется k-ой ресничкой j-го ребра глубины i, где $1 \le i \le l$, $1 \le j \le l-i$, $1 \le k \le n$, а нумерация ребер одной глубины идет слева направо. Тогда в каждый момент t конфигурацию загрузки V'(t)в І-дереве $D^{-1}(b,r,n,l;V')$ можно задать набором

$$q(t) = (q_{111}(t), q_{112}(t), \dots, q_{ijk}(t), \dots, q_{l2^{l-1}n}(t)),$$

в котором каждая координата $q_{ijk}(t)$ равна нагрузке реснички с номером ijk в момент t, причем $0 \le q_{ijk}(t) \le 2^{l-i}b$ и $\sum_{111}^{l2^{l-1}n}q_{ijk}(t) = V'(t)$. Пусть конфигурации загрузки V'(t) в каждый момент t изменяются по правилам $A_1) - E_1)$ и

 $A_2) - \mathcal{A}_2$) под воздействием квазислов из $A^* \cup A^{\omega}$.

Процесс "дыхания" можно представить некоторым инициальным конечным автоматом [3] $A_{q_0} =$ $(A,Q\cup q',\varphi,q_0)$ без выхода, где $A=\{0,1,\ldots,V\},\ Q$ - множество таких конфигураций $q=(q_{111},q_{112},\ldots,q_{ijk},\ldots,q_{l2^{l-1}n}),$ для которых выполнено $\sum_{111}^{l2^{l-1}n}q_{ijk}(t)\leq V,$ а q' - такое состояние автомата A_{q_0} , в которое он переходит под воздействием такой входной буквы a(t), что загрузка V'(t) І-дерева $D^{-1}(b,r,n,l;V')$ в момент t после подачи этой буквы стала больше объема V этого Ідерева. Изменению конфигураций загрузок V'(t) в каждый момент t по соответствующим правилам ставится в соответствие функция $\varphi: Q \times A \to Q$ переходов автомата A_{q_0} , которая доопределяется для состояния q' так: $\varphi(q',a)=q'$ для любой буквы a из A. Множеству Q_δ поставим в соответствие множество Q_{δ} таких состояний $q=(q_{111},q_{112},\ldots,q_{ijk},\ldots,q_{l2^{l-1}n})$ автомата A_{q_0} , для которых выполнено $\sum_{111}^{l2^{l-1}n} q_{ijk}(t) \leq \delta V$.

Пусть проекция вектора q на координату ijk есть q_{ijk} , то есть $pr^{ijk}q=q_{ijk}$. Обозначим через |q| сумму всех координат вектора q, то есть $|q|=\sum_{111}^{l2^{l-1}n}q_{ijk}$. Введенный автомат $A_{q_0}=(A,Q\cup\{q'\},\varphi,q_0)$, где $q_0\in Q_\delta$, назовем автоматом, ассоциированным

c І-деревом $D^{-1}(b,r,n,l;V')$.

Пусть $Q_{D^{-1}}(\delta V, b, r, n, l)$ - множество всех допустимых слов для І-дерева $D^{-1}(b, r, n, l; V')$, $T_{D^{-1}}(\delta V, b, r, n, l)$ - множество всех предельно допустимых слов для этого I-дерева, $Q_{D^{-1}}^{\omega}(\delta V, b, r, n, l)$ - множество всех допустимых сверхслов для І-дерева $D^{-1}(b,r,n,l;V')$ и $T_{D^{-1}}^{\omega}(\delta V,b,r,n,l)$ - множество всех предельно допустимых сверхслов для этого І-дерева.

Приведем вариант утверждения из [2], описывающего все множество $T_{D^{-1}}(\delta V, b, r, n, l)$, а, следовательно, и все множество $Q_{D^{-1}}(\delta V, b, r, n, l)$.

Теорема 2 Слово $\alpha^m = a(1)a(2)\dots a(t)\dots a(m)$ из A^* принадлежит $T_{D^{-1}}(\delta V,b,r,n,l)$ точно тогда, когда для него выполнены следующие рекуррентные соотношения:

а) для всех $t=1,\ldots,m-1$ имеем

$$a(t) \in \left\{ \begin{array}{l} \{0\} \cup [2^{l-1}r, \ \delta V - |q(t)|], \quad ecnu \ (pr^{111}q(t) = 0) \wedge (|q(t)| \neq 0), \\ [2^{l-1}r \dot{-} pr^{111}q(t), \ \delta V - |q(t)|], \quad unaue, \end{array} \right.$$

б) для t=m имеем $a(m)=\delta V-|q(t)|.$

Удается также на автоматном языке описать множества $Q_{D^{-1}}(\delta V, b, r, n, l), T_{D^{-1}}(\delta V, b, r, n, l),$ $Q_{D^{-1}}^{\omega}(\delta V, b, r, n, l)$ и $T_{D^{-1}}^{\omega}(\delta V, b, r, n, l)$.

Теорема 3 Множества $Q_{D^{-1}}(\delta V,b,r,n,l)$ и $T_{D^{-1}}(\delta V,b,r,n,l)$ регулярны, а множества $Q_{D^{-1}}^{\omega}(\delta V, b, r, n, l)$ и $T_{D^{-1}}^{\omega}(\delta V, b, r, n, l)$ общерегулярны.

Пусть $T_{D^{-1}}^m(\delta V,b,r,n,l)$ - множество всех предельно допустимых слов для І-дерева $D^{-1}(b,r,n,l;V')$ длины m. Обозначим мощность множества $T_{D^{-1}}^m(\delta V,b,r,n,l)$ через N(m).

Теорема 4 Для N(m) выполнены следующие соотношения:

- a) echu $\delta V \leq 2^{l-1}r$, mo N(m)=1, δ 0 echu $\delta V > 2^{l-1}r$, mo $\log_2 N(m) \asymp m$ npu $m \to \infty$.

Автор выражает благодарность академикам Кудрявцеву Валерию Борисовичу и Чучалину Александру Григорьевичу за постановку задачи и научное руководство.

- [1] Гераськина Ю. Г. Модель самоочищения легочных структур // Интеллектуальные системы 2002-2003. — Т. 7, вып. 1-4, — С. 41–54.
- [2] Гераськина Ю. Г. Модель процесса дыхания живых организмов // Интеллектуальные системы — 2003. — Т. 8, вып. 1-4, — С. 429-456.
- [3] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов // М.: Наука, 1985. - 320 с.

Автоматический синтез программ в Интеллсист

Григорьев Р. Д., Красилов А. А., e-mail: gregor@galaktika.ru Отдел интеллектуальных систем 125252, Москва, Волоколамское шоссе, 1 оф. 412

В данной статье рассматривается процесс автоматического синтеза программ в интеллектуальной системе Интеллсист. Описано решение запроса Интеллсист в виде синтезированной программы (СП), её сходства и различия с другими видами решений. Приведены примеры реализаций на языке программирования Паскаль основных конструкций языка профессиональной прозы (ЯПП). Рассмотрены типы СП и виды использования терминов в них. Даны представления отчётов по работе СП и синтезатора программ.

1. Введение в спецификацию программы

По тексту запроса, созданного пользователем на языке профессиональной прозы (ЯПП), через процессы трансляции текста, интерпретации операций, логического и алгебраического вывода, выполняемых ядром Интеллсист [1], получается результат в виде альтернатив совместных фактов. Факты могут быть:

- 1. Численным результатом или простейшими утверждениями, например, X = 1, Y = 2.
- 2. Аналитическим результатом или сложными утверждениями, например,
 - $X = \sin(A) + \cos(B).$
- 3. Программным результатом или действиями, например, P(X) или А:=В.

Конечно, результаты могут быть и ошибочными, содержать ошибки из 13 классов [2]. Результаты могут быть смешанными, содержащими численные, аналитические и программные фрагменты.

Процессы получения решений завешаются построением альтернатив (перечнем) совместных фактов (список). Если все факты имеют вид (1), то можно считать, что окончательное решение получено без синтезатора программ. Как в этом случае, так и во всех остальных можно перейти к синтезу программы. Выполнение программы должно завершаться автоматической печатью альтернатив совместных фактов типа (1).

По заказу пользователя в любом случае можно синтезировать программу (СП). Работу по синтезу выполняет подсистема Интеллсист, именуемая синтезатором. Это необходимо, если для решения запроса не хватает каких-либо данных. СП может запросить исходные данные. Кроме того, пользователь с помощью синтезатора может построить СП по заданному на ЯПП алгоритму. Для этого в языке Лейбниц, который служит для описания внутренних представлений знаний на основе записей на ЯПП, существует ряд специальных директивных операций, связанных с явным формированием потока управления. СП является программным представлением решения запроса пользователя, она состоит из подпрограмм вычисления фактов, утверждений или их частей, полученных в результате решения запроса. Поэтому СП структурно также состоит из подпрограмм альтернатив совместных фактов. В общем случае здесь каждый совместный факт может быть расчетным, нерасчетным или частично расчетным (при решении запроса каждый факт является всегда нерасчетным). Расчетные части совместных фактов альтернативы формируют альтернативные решения (алгоритмы расчета) задания, а нерасчетные - основу альтернативных результатов (для печати решения в форме альтернатив). При отсутствии в фактах расчетных частей альтернатива решения отсутствует, а альтернатива результата эквивалентна альтернативе при решении запроса (в этом случае СП ничего по альтернативе не вычисляет, а просто ее печатает). Если же, наоборот, факты состоят только из расчетных частей, то альтернатива результата будет отсутствовать (СП ничего по альтернативе не печатает, но полностью по

Факт или утверждение о факте являются расчетными, если все их операнды, операции и вызываемые подпрограммы являются полностью определенными или их определение может быть уточнено в процессе генерации СП (полнота знаний о фактах). Если же это не так, то факты разделяются на нерасчетные и расчетные части. При полной неопределенности знаний фраза является полностью нерасчетной. Кроме неполноты знаний косвенно управляет пользователь через определение списка неопределенных терминов СП.

Например, рассмотрим факт «R=(A+B) смешать с (C-D)». Если все операнды имеют конкретный тип данных, все операции определены для этих типов, и здесь нет неопределенных терминов, то весь факт является расчетным. Допустим теперь, что R – неопределенный термин, а операция «смешать с» не определена для типов выражений в скобках, тогда расчетные части – это «(A+B)» и «(C-D)», а нерасчетные части – это «R=» и «смешать с ». В результате, например, при определённых значениях A, B, C, D будет рассчитано и напечатано «R= 3 смешать с 5». Если же здесь наблюдается полная неопределенность знаний или все участвующие в выражении операнды являются неопределенными, то данный факт является полностью нерасчетным.

Выполнение СП состоит в последовательном расчете всех альтернатив решения запроса в Интеллсист. Выполнение альтернативы состоит в выполнении ее совместных фактов. Совместный факт всегда является логическим выражением. Если факт имеет значение истины, то рассчитывается следующий факт альтернативы, иначе (когда некоторый факт ложен) вся альтернатива считается ложной (противоречивой), ее выполнение прекращается, и начинается выполнение следующей альтернативы. Как было уже сказано, нерасчетные факты в СП не учитываются. Частично расчетные факты считаются всегда истинными, их расчетные части выполняются, а результаты расчета печатаются в соответствующих местах альтернативы результата. Если рассмотреть предыдущий пример частично расчетного факта и предположить, что A = 5, B = 4, C = 3, D = 2, то в альтернативе результата будет напечатано «R = 9 смешать с R = 9 смешать с R

Операции явного формирования потока управления, если используются алгоритмические записи, считаются истинными. Факты, выраженные этими операциями, могут быть ложными только при невозможности их выполнения, что приводит к прекращению работы СП вообще. В эту группу входят операции:

присваивание значения;

обмен значениями;

вызов процедуры;

переход на метку:

все логические операции, где хотя бы один из операндов является результатом перечисленных операций;

катенация фактов;

Смысл этих операций интуитивно понятен, стоит только пояснить операцию катенации фактов. Эта операция задает однозначный порядок выполнения последовательности фактов. Иными словами операция катенации - это возможность пользователя указать Интеллсист, что ему важен порядок выполнения выражений. Строго говоря, эти операции могут считаться имеющими также и ложный результат. Это возможно, если в результате возникшей исключительной ситуации (и только по этой причине) факт не выполнился, но система обработки исключений передала управление не на выход из СП, а на продолжение выполнения СП со следующей операции (или других мест в общем случае). Например, операция «Р1 или Р2» будет считаться ложной при возникновении исключений в обеих процедурах. Данная методика требует поддержки со стороны ЯПП и находится в настоящее время в разработке.

Кроме того, здесь следует заметить о правилах выполнения операндов бинарных логических операций – как логического, так и процедурного типов. Можно применять правило невыполнения второго операнда, если результат первого операнда однозначно определяет значение результата операции. Например, в операции «А и В», если А – ложно, то В можно не вычислять – результат будет обязательно ложным. Можно применять противоположное правило – обязательно вычислять оба операнда. Особую сложность здесь вызывает реализация вычислений процедурных операндов в логических операциях. Если записано «А и Р» (А – логическое, Р - процедурное), то надо ли вычислять А? Более того, может не надо исполнять Р – результат будет всегда истинным? Как было сказано выше, при возникновении исключения в Р результат станет ложным, и если он может использоваться в дальнейших вычислениях, то Р должно вычисляться. Оптимальным решением здесь представляется обеспечение пользователю прагматического управления применением этих правил. В текущей реализации синтезатора программ для логических операндов правила вычисления определяются языком программирования, а для процедурных и смешанных операндов используется правило частичного вычисления. Ниже в таблице перечислены реализации логических операций в настоящее время (а, b – логические, p, q – процедурные операнды). Здесь необходимо учесть, что ряд бинарных логических операций являются коммутативными.

ППК	Паскаль
a:=b & p;	p; a:=b {and TRUE};
a:=(p & q);	p; {if TRUE then }q; a:=TRUE {and TRUE};
$a:=(b \lor p);$	if not b then p; a:=TRUE {or b};
$a := (p \lor q);$	p; {if FALSE then q;} a:=TRUE {or TRUE};
a:=(b xor p);	p; a:={TRUE xor b =} not b;
a:=p xor q;	p; q; a:={TRUE xor TRUE =}FALSE;

a:=b -> p;	if b then p;
	$a:=\{not\ b\ or\ TRUE=\}\ TRUE;$
a:=p -> b;	p;
	$a:=\{\text{not TRUE or } b=\} b;$
a:=p -> q;	p;
	{if not FALSE then }q;
	a:={not TRUE or TRUE =} TRUE;
a:=not p;	p;
	a:={not TRUE = }FALSE;

2. Синтез программ по открытым и закрытым фразам

Запрос пользователя рассматривается как логическое выражение исчисления предикатов первого (сейчас) или высокого порядков. Квантор всеобщности для переменной понимается как «дано», а квантор существования – как «найти». Если выражение хотя бы по одной переменной не охвачен какимлибо квантором, оно считается открытым, в противном случае – закрытым. Результатом решения открытого запроса будет аналитическое выражение для связной переменной квантора найти (скорее всего). Результатом решения закрытого запроса будет числовое выражение для связной переменной квантора найти (скорее всего). Квантор дано используется для задания исходных данных.

Конечно, в большинстве случаев запросы представляют открытые выражения. Синтезатор программ запрашивает квалификацию величин. Они могут быть исходными, искомыми, расчетными, неопределенными.

Квалификация осуществляется пользователем в зависимости от постановки задачи, выраженной запросом. Смысл исходных и искомых терминов понятен из названий. При этом не все термины могут быть определены как исходные или искомые. Например, это функции, процедуры, ссылочные термины, т.к. их значением является последовательность байтов или число, смысл которых для пользователя является абстрактным.

Расчетные термины являются рабочими величинами, хранилищами промежуточных результатов. Они получают значения от одних терминов и передают их другим в процессе исполнения СП.

Неопределенные термины формируют нерасчетные части выражения и своими именами идут в альтернативы результата. Например, если в выражении X = Y + Z, где X – неопределенный, а Y равен 6, и Z равен 5, то пользователь увидит решение в виде X = 11.

Квалификация величин полезна в двух случаях. Квалификация величин определяет практически полезные прямую и обратную задачи. Пусть, например, поставлена задача A(x,y)=0. Прямая задача получается при квалификации x - (запрос закрыт квантором «дано») y - (запрос закрыт квантором «найти») и построении решения в виде $y=\Pi(x)$. Обратная задача получается при квалификации x - (запрос закрыт квантором «найти») y - (запрос закрыт квантором «дано») и построении решения в виде x=O(y). Второй случай пользы квалификации величин состоит в ориентации подпрограмм аналитического вывода на разрешение логических уравнений относительно связных переменных квантора «найти».

3. Подстановки при синтезе

Как известно каждая программа содержит в себе линейные участки исполнения, ветвления и циклы. СП здесь не исключение. Ветвления и циклы формируются в СП как явно командами пользователя, так и опосредовано логикой исполнения.

Ветвление и циклы явно задаются операцией перехода на метку как самой по себе (безусловный переход) так и в сочетании с логической операцией, как правило – если-то (условный переход).

Конструкция	ппк	Паскаль
Описание термина	раздел понятий	Var
	\sim масса \sim : $mun := значение;$	Z245Z: тип = значение;
Описание типа	раздел типов	Туре
	~Oм~ = значение;	Z1245Z = значение;
Описание	раздел синонимов	Const
синонима	~сторожить~ =	Z2245Z = Z2224Z;
	~караулить~;	{только для ординарных константных терминов}
Описание	раздел операций	{нет}
пользовательской	~+~(операнды) =	
операции	значение;	
(перегрузка		
встроенной)		
Описание	раздел операций	{нет}
пользовательской	~смешать	

операции (новой)	с~(операнды) = значение;	
Кванторное	Bce ~i~: mun	Var
выражение «для	(выражение(i));	Z345Z: <i>mun</i> ;
BCCX>>		Z1VZ: Boolean;
		Z345Z:=первое значение;
		Z1VZ:=TRUE;
		while TRUE do begin
		Z1VZ:=выражение(Z345Z);
		if not Z1VZ then Break;
		if $Z345Z = nocnednee$ значение then Break;
		Inc(Z345Z);
		end;
Кванторное	существует ~і~: тип	Var
выражение	(выражение(і));	Z345Z: <i>mun</i> ;
«существует»	(Z1VZ: Boolean;
		Z345Z:=первое значение;
		Z1VZ:=FALSE;
		while TRUE do begin
		Z1VZ:=выражение(Z345Z);
		if Z1VZ then Break;
		if $Z345Z = nocnedhee$ значение then Break;
		Inc(Z345Z);
		end;
Описание	Раздел меток	Label Z445Z;
меток	~цикл по списку~;	24047 2 1 1024
Переход на	перейти к циклу по	Goto Z445Z;
метку	списку;	0000 21132,
Присваивание;	a:=b;	Z245Z:=Z246Z;
Обратное	a=:b;	Z246Z:=Z245Z;
присваивание;	a:=:b;	{HeT}
Обмен	,	()
Катенация	A^; B^; C^.	Z145Z; Z245Z; Z345Z;
фактов		
Ветвление	если дождь, то взять	if Z445Z then Z446Z;
	зонтик;	
Цикл	начало цикла:	Z123Z:
	если надо	if Z124Z then goto Z123Z;
	продолжать, то перейти к	
	началу цикла;	
	in mily minim,	

СП. Синтез производится с помощью набора файлов-шаблонов на операции ЯПП для каждого языка программирования.

В процессе синтеза программы необходимо порождаются идентификаторы для обозначения терминов в СП, рабочих ячеек, имен альтернатив, новых и предопределенных объектов, имен программных модулей. Так как термин может состоять из нескольких слов, лексем и символов, отличных от принятых для идентификаторов в большинстве языков программирования, то он не может быть использован в этом качестве. Таблица идентификаторов представлена ниже вместе с сопоставлением с терминами.

Идентификатор	Назначение	Пример
ZnnZ	nn – номер термина в лексиконе	Z345Z – для термина «скорость
		объекта»
ZnnSZ	nn – номер стандартного элемента СП	Z126SZ – модуль стандартных
		величин
ZnnTZ	nn – номер сгенерированного типа	Z1TZ – для типа, описывающего
	данных в СП	агрегат записи, используемого в СП
ZnnFZ	nn – номер сгенерированного	Z12FZ – для входящего в запись поля,
	вложенного элемента в СП	которая описывающего элемент
		используемого в СП агрегата записи
ZnnVZ	nn – номер сгенерированного элемента	Z15VZ – новая величина
	в СП (рабочей ячейки)	

ZnnPZ	nn – номер элемента СП,	Z345PZ – входной параметр СПП со
	соответствующего термину в лексиконе с	значением для термина Z345SZ «скорость
	таким же номером, но не описывающего	объекта»
	данный термин в СП (для этого есть ZnnZ)	
ZnnAZ	Имя процедура для альтернативы	Z5AZ – пятая альтернатива
Идентификатор	Идентификатор в языке	Sin, sqrt, random,
любого вида	программирования (внешний термин или	
	системная величина)	

4. Назначения СП

СП могут различаться по трем категориям исполнения: реализация, вариант, интерфейс. Реализацией или видом использования СП являются:

- 1. Программа автономная задача, которая генерируется в виде исполняемого файла (ЕХЕ).
- 2. Исполняемая подпрограмма вид СП (второй сорт реализации), которая выполняет какую-либо отдельную функцию в рамках подзадачи некоторой большой задачи. Она генерируется в файле библиотеки динамической компоновки (.DLL). Такой вид подпрограмм позволяет строить пакеты прикладных программ.
- 3. Компонуемая подпрограмма вид СП (третий сорт реализации), которая генерируется в файлах исходного кода на языке программирования или промежуточного кода. СП в данной реализации также выполняет какую-либо отдельную функцию в рамках подзадачи. Однако, такая СП не готова к использованию, так как необходима компиляция файлов СП в исполняемый код и/или связывание кода СП с кодом внешней программы, для которой она создается. Такая реализация может обеспечить большую гибкость для пользователей-программистов.

Вариант СП определяет принцип ее использования, главными являются следующие:

- 1. Отладочный вариант. Он обладает свойствами тщательного контроля над правильностью счета и содержит обработчики стандартных ошибок и исключительных ситуаций, например, деление на ноль. В тело СП вставляются процедуры определения контекста выполнения операций, что позволяет пользователю относительно быстро найти место возникновения ошибки или исключительной ситуации.
- 2. Счетный вариант. Он обладает меньшим контролем над правильностью счета, что значительно ускоряет процесс выполнения СП. Счетный вариант подразумевает обязательное наличие потока результатов.
- 3. Встраиваемый вариант. Потоки исходных данных и результата подразумевают обмен данными по каналам связи с технической системой (обмен информацией с внешними устройствами) с помощью процедур приема и выдачи данных. В конечном итоге результат может получить и пользователь, например, при применении пользовательского интерфейса СП. Для таких программ существенны КВ «Интерфейс» и «Прерывание».

Интерфейс определяет способ обмена информацией между пользователем и СП и подразделяется на:

- 1. Пользовательский интерфейс. Решение задачи обмена информацией с СП полностью ложится на пользователя. С помощью КВ «Интерфейс» пользователь пишет тексты заданий для определения того, как и когда будет осуществляться обмен информацией между ним и его программой. Необходимость знания содержимого КВ «Интерфейс» компенсируется значительными гибкостью и мощностью обмена.
- 2. Стандартный интерфейс. Обмен информацией между пользователем и СП производится через командную строку запуска СП по определенному и известному формату.
- 3. Стандартный оконный интерфейс. Обмен информацией между пользователем и СП осуществляется через графическую систему окон, меню и элементов управления, подключаемую Интеллсист.

5. Итоговые печати синтеза программ

В общем случае работа пользователя с СП сводится к заданию исходных данных, автоматическому выполнению расчета, получению результата (в том числе и противоречивого). Результатами выполнения СП являются альтернативы решений, сообщения о противоречиях и ошибках. Задание может состоять из несколько вариантов значений исходных данных, которые могут храниться в файле. Стандартные интерфейсы позволяют: загружать (сохранять) варианты исходных данных из файла данных, редактировать и контролировать значения исходных терминов, создавать новые варианты заданий и удалять существующие, задавать произвольную последовательность исполнения вариантов, запускать варианты на счет, получать результат расчета, противоречия и сообщения об ошибках в отдельных потоках, сохранять результаты в файлы. Результаты сохраняются раздельно по трем файлам – положительные результаты, противоречия, сообщения об ошибках. Отпечатанные результаты выглядят аналогично тому, как они представляются после разрешения запроса без построения СП.

Синтез программы завершается печатью отчета о С Π – это есть готовый текстовый файл. В генерируемый отчет для С Π входят:

- имя СП;
- дата и время создания;
- сведения об авторе и организации;
- версия Интеллсист;
- описание интерфейса доступа к СПП в реализации исполняемой подпрограммы;
- вид СП (реализация, вариант, интерфейс);
- формальный язык, на котором создана СП;
- каталог создания СП;
- список модулей СП с указанием их назначения, числа строчек и размера;
- список остальных файлов СП с указанием их назначения и размера;
- список требуемых внешних, системных и интерфейсных модулей;
- список идентификаторов используемых БД;
- список таблиц с указанием их идентификаторов, идентификаторов БД, вида использования;
- статистика по СП: число элементов по разным категориям и приблизительные размеры различных областей памяти, занимаемых СП;
- список терминов СП с указанием идентификатора и вида использования.

В генерируемый отчет по ППП в целом входит:

- имя ППП:
- дата и время последнего изменения;
- сведения об авторе и организации;
- версия Интеллсист;
- вид ППП (реализация и вариант);
- каталог создания ППП;
- список имен СПП, входящих в ППП.

Формат печати результата работы СП является подобным формату печати решения запроса, так как результатом и в том и в другом случае являются альтернативы совместных фактов. Для идентификации решения в «пространстве» здесь печатается название запроса, по которому была построена СП, и номер варианта исходных данных, по которому проводился расчет. Также для идентификации решения во «времени» печатается дата и время расчета.

Кроме печати альтернатив совместных фактов в результате печатается различная дополнительная информация, которая разделяется на общую и альтернативную. Общая дополнительная информация является той же дополнительной информацией, что и в решении запроса — это, прежде всего различные предупреждения, например: «Необходимы знания о выведенных величинах и операциях!»

Печать результата альтернативы состоит из печати совместных фактов и возможной печати списка неопределенных функций, процедур и операций, вызываемых из альтернативы, и списка искомых терминов с их значениями.

Печать совместных фактов представляет собой совместные факты, разделенные запятой. Печать совместного факта в общем случае состоит из нерасчетной (печатаемой) последовательности символов, внутри которой находится различное число расчетных «окон». В этих «окнах» производится печать соответствующих значений в соответствии с их типом данных. Факт может состоять из одной нерасчетной части, и тогда его печать в СП не отличается от печати в решении запроса. Если же у факта нет вообще нерасчетной части (т.е. он состоит только из одной расчетной части), то он исключается из печати альтернативы, и вместо этого управляет альтернативой или выполняет какое-либо алгоритмическое действие.

При отсутствии альтернатив результата, например, когда все факты всех альтернатив состоят только из расчетных частей, на месте альтернатив печатается служебное сообщение, вид которого зависит от наличия глобальных или локальных противоречий.

Заметим, что пользователь не применяет процедуры ввода и вывода!

- 1. <u>www.intellsyst.ru</u>
- 2. Красилов А. А Экология информации и знаний. Научно-техническая информация, серия 1, Организация и методика информационной работы. № 2. М., ВИНИТИ, 2005, с. 11-23

Метод построения расписания для задачи минимизации максимального запаздывания

Григорьева Н. С.,

С.-Петербургский гос. университет, доцент 195279 С.-Петербург пр. Ударников 38-2-173 E-mail: gns1175@ztel.spb.ru

Рассматривается задача минимизации максимального запаздывания для параллельных процессоров с ограничениями на порядок выполнения работ. Эта задача относится к задачам составления расписаний, имеет многочисленные приложения и является NP-трудной [1]. Для решения задачи предлагается метод ветвей и границ, который позволяет получать точное оптимальное решение или, при ограничении на время работы алгоритма, достаточно хорошие приближенные решения задачи. Основными особенностями данного метода являются метод продолжения частичного решения и метод его оценки.

Предлагается метод продолжения частичного решения, позволяющий включать невынужденные простои в расписание выполнения заданий. Разрешение невынужденных простоев может приводить к более быстрому получению оптимального решения. В [2] содержится обзор методов составления расписаний с простоями, но для рассматриваемой задачи такой подход ранее не применялся.

Для оценки частичного решения, предлагается метод, основанный на определении интервала концентрации. Был проведен вычислительный эксперимент, который показал эффективность предложенных подходов.

Рассмотрим систему заданий $U = \{u_1, u_2, \ldots, u_n\}$, на которых задано отношение частичного порядка \prec , выражение $u_i \prec u_j$ означает, что выполнение задания u_j может быть начато только после завершения задания u_i . Будем считать, что отношение частичного порядка задано в виде графа $G = \langle U, E \rangle$, в котором есть дуга $e = (u_i, u_j) \in E$ тогда и только тогда, когда $u_i \prec u_j$. Задано время выполнения каждого задания $t(u_i)$ и директивный срок $d(u_i)$, к которому задание должно быть закончено.

Для выполнения заданий имеется m идентичных процессоров, любое задание может выполняться на любом процессоре, и каждый процессор может выполнять не более одного задания в каждый момент времени. Прерывания выполнения заданий не допускаются.

Построить расписание значит найти для каждого задания u_i время начала выполнения задания $\tau(u_i)$ и номер процессора $num(u_i)$, на котором оно выполняется. Определим запаздывание для каждого задания $f(u_i) = \max\{0, \tau(u_i) + t(u_i) - d(u_i)\}$. Назовем максимальным запаздыванием для расписания S величину $L_S = \max\{f(u_i)|u_i \in U\}$. Требуется построить расписание S минимизирующее максимальное запаздывание.

Метод ветвей и границ построения оптимального расписания

Расписание будем рассматривать как перестановку заданий. По каждой перестановке заданий $\pi = (u_1, u_2, \ldots, u_n)$, можно построить расписание S_{π} следующим образом: очередное задание назначается на процессор, который раньше всего освободился, в самый ранний возможный момент времени, тогда каждая перестановка будет однозначно определять расписание S_{π} .

Частичное решение σ_k , где k число заданий, будем рассматривать как частичную перестановку $\sigma_k = (u_1, u_2, \dots, u_k)$, по которой однозначно определяется частичное расписание.

Для каждого задания u определим раннее время начала выполнения задания $v_{min}(u)$, которое численно равно длине максимального пути в графе G от начальной вершины до задания u. Ранние начала по ходу составления расписания будут увеличиваться, т.к. назначения заданий на процессоры будут приводить к новым отношениям порядка на множестве заданий. Для заданий u_i , входящих в частичное решение σ_k , известно время начала выполнения заданий $\tau(u_i)$ и номер процессора $num(u_i)$, на котором оно выполняется, $r(u_i)$ — время простоя процессора перед началом выполнения задания u_i . Известно время освобождения процессоров $time_k[1:m]$. Обозначим $t_{min}(k) = \min\{time_k[i]|i \in 1:m\}$ время, в которое освобождается самый ранний процессор. Известен рекорд rec— значение целевой функции, для лучшего на данный момент решения и оценка частичного решения $LB(\sigma_k)$.

Правило выбора задания имеет большое значение в методе ветвей и границ. В списочных алгоритмах готовым в данный момент времени называется задание, все предшественники которого закончены к этому моменту. Введем другое определение готового задания.

Определение. Задание u будем называть готовым на уровне перебора k, если все его предшественники включены в частичное решение σ_{k-1} .

Такое определение готового задания отличается от обычного определения и позволяет перебрать все возможные варианты составления расписаний, включая те, в которых процессор простаивает при наличии готовых заданий. Такие простои будем называть невынужденными простоями.

Будем использовать левосторонний обход дерева вариантов и помнить R_k — множество заданий, запрещенных на уровне перебора k, т.е. тех заданий, которые были готовыми заданиями, были проверены и отброшены.

Задания готовые и не запрещенные на уровне перебора k будем называть допустимыми заданиями на уровне k и множество таких заданий обозначать U_k^* .

Важным параметром метода ветвей и границ является способ выбора задания из множества допустимых. Предлагается сначала выбирають задания с минимальным директивным сроком, при этом самые критические задания будут назначаться в первую очередь. Этот вариант может приводить к частому появлению простоев, чтобы исключить нерациональное использование времени процессора, будем подбирать задание, которое можно выполнять в этот период простоя. Поэтому выбор задания будет осуществляться в два этапа. Такой метод выбора заданий был разработан для задачи построения расписания минимальной длины [3].

Процедура выбора задания $Select(U_k^*, t_{min}(k), u_0)$

Из множества допустимых заданий U_k^* выбирается задание u_0 с минимальным директивным сроком. Если перед выполнением этого задания возникает простой процессора, то пытаемся подобрать задание, которое можно выполнять в период простоя.

1. Выберем задание u_0 , для которого выполнено

$$d(u_0) = \min\{d(u_i), | u_i \in U_k^*\}.$$

- 2. Определим величину $r_0 = v_{min}(u_0) t_{min}(k)$ простоя процессора l_0 перед выполнением задания u_0 .
- 3. Если $r_0 > 0$, то выберем из множества допустимых заданий такое задание u^* , которое можно будет выполнить в период простоя процессора без увеличения времени начала задания u_0 , а именно

$$\max\{t_{min}(k), v_{min}(u^*)\} + t(u^*) \le v_{min}(u_0).$$

4. Если такое задание нашлось, то будем назначать на процессор задание u^* , т.е. положим $u_0 := u^*$, в противном случае будем назначать задание u_0 .

Вторым важным параметром метода ветвей и границ является способ построения оценки частичного решения. Нижняя оценка целевой функции $LB(\sigma_k)$ для данной задачи является максимумом из двух оценок. Первая оценка $LB1(\sigma_k)$ связана с отношением порядка на множестве работ. Вторая $LB2(\sigma_k)$ больше зависит от времен выполнения заданий и директивных сроков. Для получениея этой оценки на каждом уровне перебора будем оценивать для всех интервалов времени количество требуемых ресурсов и сравнивать с количеством наличных ресурсов. Для этого предлагается модифицировать идею алгоритма определения интервала концентрации [4] для полного расписания таким образом, чтобы применить для построения оценки частичного расписания σ_k .

Процедура построения оценки $Est(U, \sigma_k; LB(\sigma_k))$

- 1. Определим значение целевой функции для частичного решения $L = \max\{\tau(u_i) + t(u_i) d(u_i) | u_i \in \sigma_k\}$
- 2. Найдем оценку целевой функции для заданий, не включенных в частичное решение $est = \max\{v_{min}(u_i) + t(u_i) d(u_i) | u_i \notin \sigma_k\}$
- 3. Определим первую оценку частичного решения $LB1(\sigma_k) = \max\{L, est, 0\}$
- 4. Будем рассматривать интервалы времени $[t_1, t_2] \subseteq [t_{min}(k), d_{max}]$. Для каждого интервала определяем общую мощность процессоров, которые свободны в данном интервале

$$MP(t_1, t_2) = \sum_{i=1}^{m} \max\{0, (t_2 - \max\{t_1, time_k[i]\})\}.$$

5. Для каждого задания $u \notin \sigma_k$, вычисляем минимальное возможное время начала выполнения задания: $v(u_i) = \max\{v_{min}(u_i), t_{min}(k)\}$ и

$$x_k(u_i) = |[v(u_i), v(u_i) + t(u_i)] \cap [t_1, t_2]|,$$

$$y(u_i) = |[d(u_i) - t(u_i), d(u_i)] \cap [t_1, t_2]|,$$

6. Определим $M_k(t_1, t_2)$ — суммарную минимальную потребность заданий в ресурсах на интервале $[t_1, t_2]$,

$$M_k(t_1, t_2) = \sum_{u_i \notin \sigma_k} \min\{x_k(u_i), y(u_i)\},$$

7. Положим

$$LB2(\sigma_k) = \max_{[t_1, t_2] \in [t_{min}(k), d_{max}]} \left(\left\lceil \frac{M_k(t_1, t_2) - MP(t_1, t_2)}{m} \right\rceil \right)$$

Верно следующее утверждение.

Утверждение Нижняя оценка целевой функции для частичного решения σ_k равна $LB(\sigma_k) = \max\{LB1(\sigma_k), LB2(\sigma_k)\}.$

Алгоритм построения оптимального расписания

Начальный шаг: k := 1; time[i] = 0; $\sigma_0 = \emptyset$;

- 1. Найдем процессор l_0 с минимальным временем освобождения $t_{min}(k)$.
- 2. Определим множество допустимых заданий на данном уровне перебора U_k^* . Если множество U_k^* пусто, то перейдем к п.6.
- 3. Вычислим оценку частичного решения процедурой $Est(U, \sigma_k; LB(\sigma_k))$, если $LB(\sigma_k) > rec$, то переходим к п.7.
- 4. Выберем задание u_0 , процедурой выбора задания: $Select(U_k^*; u_0)$
- 5. Шаг вперед. Добавим задание u_0 к частичному решению σ_{k-1} получим решение σ_k . Если k=n, то сохраним новый рекорд $rec=L(\sigma_k)$ и перейдем к п.7, иначе перейдем к п.1.
- 6. Если k>1,то аннулируем список заданий,запрещенных на уровне перебора $k,\ R_k:=\emptyset,$ иначе конеп.
- 7. Если k > 1, то делаем шаг назад, отменяя проследнее назначенное задание, и переходим к п.1., иначе конец.

- 1. Brucker P., Garey M.R. and Jonson D.S. Sheduling equal-length tasks under treelike precedence constraints ti minimize maximum lateness.// Math.Oper.Res.,1977, vol. 2,pp.275-284.
- 2. Kanet J.J.,Sridharan V. Scheduling with inserted idle time: problem taxonomy and literature review.// Operetions Research 2000,vol.48,No 1,pp.99-110.
- 3. Григорьева Н.С. Эвристический алгоритм построения расписания для многопроцессорной системы.// Тезисы докладов конференции Дискретная математика и приложения 2001, Москва,с. 99-110.
- 4. Кукса А.И. Сравнение нижних границ длительностей детерминированных мультипроцессорных расписаний.// Кибернетика, 1979 т. 5, с. 87-90.

О моделировании биологических систем (на основе поисково-оптимизационного подхода)

Гринченко С. Н.,

д.т.н., проф., ведущий научный сотрудник Института проблем информатики РАН 119333, Москва, Вавилова, д.44, кор.2 e-mail: SGrinchenko@ipiran.ru

При моделировании биологических систем предлагается рассматривать живую природу как иерархическую систему, включающую в себя механизм иерархической адаптивной поисковой оптимизации (целевых критериев энергетического характера). Даются интерпретации некоторых результатов такого представления, в частности — синтеза эволюционных позиций дарвинизма и номогенеза. Подробности предлагаемого подхода изложены в монографии автора «Системная память живого (как основа его метаэволюции и периодической структуры)» (М.: ИПИРАН, Мир, 2004, 512 с.).

Ключевые слова: моделирование, биосистемы, биологическая эволюция, кибернетика, поисковая оптимизация, экстремальное управление, случайный поиск, положительная обратная связь, иерархия, адаптация.

Моделирование биологических систем и кибернетика

Исследования в области проблемы возникновения и дальнейшего развития живого, которые проводятся биологами самых различных специализаций, базируются, как правило, на физических и химических языках и концептуальных представлениях. То есть в биологии неявно принята парадигма (исходная концептуальная схема) использования для объяснения этой проблемы языков и терминологии физики и химии. На этом фоне имеющиеся предложения о её расширении за счет привлечения языка кибернетики (см., напр., основополагающие работы И. И. Шмальгаузена, фон Берталанфи и А. А. Ляпунова, фундаментальный труд В. Ф. Турчина [1] и др.) до сих пор не получают достаточного распространения.

Но можно ли описать с необходимой полнотой понятие и проявления живого с помощью понятийного аппарата только физики и химии? Думается, что нет. Не отрицая продуктивности подобного физикалистского подхода, следует отметить, что в ряд авторов (В. С. Степин, А. Т. Шаталов, Ю. В. Олейников, С. В. Мейен и др.) ставят вопрос о коренном пересмотре прежней физикалистской (механистической) картины мира. Например, высказывается мнение, что «...физикалистские, а затем и логико-кибернетические концепции оказывались необходимыми, но не достаточными для понимания объективных особенностей генетических систем» [2]. Таким образом, назрела необходимость существенного расширения спектра языков, применяемых для интерпретации представления о живом.

По моему мнению, для этого необходимо расширить существующую парадигму, прежде всего, за счет привлечения в рассмотрение междисциплинарных областей знания, а именно кибернетики и теории сложных систем. Подобные попытки уже предпринимаются, но в фазе создания моделей (что требует привлечения специальных знаний из областей математики, конкретных направлений технической кибернетики и т.п.) по определению могут быть реализованы лишь специалистами из указанных областей знания, т.е. небиологами. По-видимому, вследствие этого они пока и не достигают своей цели – не воспринимаются большинством биологов как абсолютно необходимый базис для эффективного продвижения их собственных исследований, и не формируют новую биолого-кибернетическую (биолого-системную) парадигму представлений о возникновении жизни. Как результат, подобная терминология вообще вытесняется из обращения в данной области исследования.

Более того, в различных прикладных науках в последние десятилетия бытует предубеждение к термину «кибернетика» и содержательному его наполнению. В лучшем случае говорят о достижении на этой базе устойчивых состояний в объекте, гомеостаза и т.п. Тем самым кибернетика (наука об управлении), фактически, используется лишь как средство изучения способов введения в некоторый контур регулирования отрица-

тельной обратной связи. От подобного неверного представления о *кибернетических системах* следует отказаться, поскольку предмет и возможности кибернетики выходят далеко за эти рамки.

Прежде всего, при рассмотрении биологических систем необходимо учитывать наличие в них и *положительных* обратных связей (что обеспечивает их *развитие*). На это указывают многие авторы. Например, Н. Н. Моисеев утверждает, вслед за А. А. Богдановым, что «для развития организационной структуры, будь то социальная или биологическая, необходимы не только отрицательные, но и положительные связи» [3]. В свою очередь, Ф. Хейлиген и К. Джослин отмечают: «Многие из основных идей кибернетики были ассимилированы другими дисциплинами, где они продолжают влиять на научные разработки. Другие важные кибернетические принципы, казалось, имевшие все шансы быть забытыми, тем не менее, периодически переоткрываются или "переизобретаются" в различных областях. Примеры – возрождение нервных сетей, изобретенных кибернетиками сначала в 1940-х, затем в конце 1960-х и снова в конце 1980-х; переоткрытие важности автономного взаимодействия в областях робототехники и искусственного интеллекта в 1990-х; и значение эффекта положительной обратной связи в сложных системах, переоткрытого экономистами в 1990-х (курсив мой – С.Г.)» [4].

Но само по себе использование положительной обратной связи в кибернетических системах способно достаточно быстро приводить к возникновению лавинообразного (взрывообразного) роста характеристик соответствующих процессов в них. То есть их введение не является панацеей от всех трудностей, и должно сопровождаться одновременным введением и отрицательной обратной связи (что обеспечивает устойчивость).

Именно подобный подход фактически и был реализован в 1960-1970-х гг. рядом отечественных и зарубежных ученых в области *такой кибернетики*, которыми был создан, а в последующие годы – развит такой важный её раздел, как теория *поисковой оптимизации* (экстремального управления): это Д. И. Батищев, Р. Буш, Ч. С. Дрейпер, Ю. М. Ермольев, А. Г. Ивахненко, В. В. Казакевич, А. А. Красовский, В. М. Кунцевич, И. Т. Ли, Ф. Мостеллер, Ю. И. Неймарк, А. А. Первозванский, Л. А. Растригин, Р. Г. Стронгин, А. А. Фельдбаум, Я. З. Цыпкин, У. Р. Эшби, Д. Б. Юдин и др.

К сожалению, теория поисковой оптимизации также не приобрела широкой популярности вне области технических применений. А ведь её отличительная особенность состоит как раз в том, что, в терминах кибернетической обратной связи, поиск в контуре регулирования — это попеременная смена её знака с положительного на отрицательный и обратно. Тем самым тенденция развития (неустойчивости, расходимости еtc.), реализуемая положительной обратной связью, и тенденция сохранения (устойчивости, сходимости еtc.), реализуемая отрицательной обратной связью, попеременно сменяют друг друга. Вопрос конкретной реализации и выбора текущих параметров контура поискового управления — в том, как долго превалирует та или иная тенденция. То есть, каковы длительности и интенсивности этих режимов в соответствующих условиях функционирования системы управления. А также и в том, кто (или что) и каким образом переключает указанные режимы, т.е. какими средствами реализуется блок вычисления целевой функции управления.

При собственно поисковой оптимизации существует задача определения момента, когда поиск следует завершить. Поскольку положение экстремума целевой функции а priori неизвестно, определять этот момент с помощью задания величины допустимого отклонения от цели невозможно. Но это можно делать по достижению числом последовательных неудачных поисковых шагов (или проб) некоторого заранее определенного предела. Последнее можно интерпретировать как признак того, что поиск производится вблизи экстремального значения целевого критерия.

Указанная задача отсутствует в системах экстремального управления, отслеживающих экстремум оптимизируемого критерия перманентно. Но здесь возникает новая проблема выбора наилучшего соотношения между двумя противоречивыми требованиями: минимизации времени переходного процесса в область экстремума из произвольной точки (так называемых «потерь на поиск») и минимизации отклонения оптимизируемой величины от экстремального значения в установившемся режиме (так называемых «потерь на рысканье»). Потери на поиск уменьшаются с ростом величины поискового шага. Однако потери на поиск характеризуют работу алгоритма лишь на подходе к экстремуму. Алгоритм, имеющий наименьшие потери на поиск, быстро выведет систему в район экстремума. Но на этом не заканчивается экстремальное управление: от алгоритмов поиска требуется надежная работа в районе экстремума, позволяющая отслеживать его «блуждание» или «уплывание». Потери на рысканье существенно зависят от величины поисковых шагов, поэтому для снижения этих потерь размеры поисковых шагов следует уменьшать. Это находится в противоречии с требованием быстродействия (для повышения которого величину рабочего шага нужно, наоборот, увеличивать).

Отсюда следует, что выбор компромиссных значений величин шагов на всей траектории поиска следует проводить *адаптивно*, с учетом особенностей поведения оптимизируемого объекта в процессе поиска [5].

Среди направлений дальнейшего развития работ в области поисковой оптимизации важное место занимают так называемые бионические алгоритмы, базирующиеся на информацию о структуре и поведении различных биологических объектов. Понятно, что инженеры-разработчики используют при этом не «истинные» свойства таких объектов, а те или иные особенности современного модельного представления о них. Чаще это происходит в упрощенной форме «биологических метафор», которые применяются в тех же целях и в других научных областях, в частности, в социологии. Среди разнообразных бионических алгоритмов поисковой оптимизации следует упомянуть «эволюционный подход» (Г. Бремерманн), «эволюционное моделирование» (Л. Фогель, А. Оуэнс, М. Уоліп; И. Л. Букатова с соавт.), «эволюционное формообразование» (А. И. Половинкин), «эволюционную глобальную оптимизацию» (Э. М. Куссуль, А. Лук; Р. Джарвис), «эволюционную адаптацию коллективом вероятностных автоматов» (Ю. И. Неймарк с соавт.), «эволюционную стратегию» (И. Рехенберг, Г.-П. Швефель, К. Беллман, Дж. Борн), «генетическую адаптацию» (У. Петерзон, К. Восс, К. Вебер), «генетические алгоритмы» (Дж. Холланд, Д. Голдберг, Д. И. Батищев, В. М. Курейчик, С. А. Исаев и др.) и тому подобные. Использование именно эволюционной и генетической терминологии оказалось удобным и достаточно эффективным языком при синтезе новых разновидностей алгоритмов поисковой оптимизации. Среди авторов бионических работ в области поисковой оптимизации, опирающихся на аналогии с иными биологическими объектами, существенный вклад внесли В. Ф. Короп, Л. Н. Фицнер, Л. А. Растригин, Н. П. Дидиченко и др. (подробности и ссылки на литературу по этой тематике см. в [6]).

Здесь необходимо отметить, что факт проведения столь большого числа достаточно успешных инженерных разработок в области поисковой оптимизации, имеющих в своей основе пусть и крайне упрощенные, но все же модели биологических системных процессов, представляется весьма показательным. По-видимому, некая глубинная связь идеологий этих феноменов – кибернетического поискового оптимизационного механизма и адаптивных механизмов живого – существует объективно. Но в рамках указанных вполне прагматических, технико-ориентированных разработок, задача построения адекватных моделей собственно биологических явлений просто не ставится. Следовательно, опираясь на отмеченную их внутреннюю связь, становится актуальной именно такая постановка вопроса: встает задача синтеза модели некоторого биоявления/биопроцесса на базе механизмов поисковой оптимизации. И первым биоявлением в этом ряду, несомненно, является приспособительное поведение живых систем.

Все это перекликается и с недавно появившимися в философской литературе предложением (восходящим к Анаксагору и Платону и базирующимся на огромном материале последующих поколений исследователей всех отраслей знания) о конституировании новой трансдисциплинарной науки — оптимологии [7]. Автор этого предложения О.С.Разумовский отмечает, что «исходной предпосылкой оптимологии должен быть эмпирический факт: главный вектор существования, организации, функционирования и развития всех живых систем, общества и человека, а также смешанных или гибридных систем (...) — интенция к оптимизации (к оптимуму)» [там же]. При этом в качестве одной из самых трудных методологических проблем он рассматривает проблему выбора прототипа для идеального объекта оптимологии, без решения которой нельзя «вообще конструировать понятийно-номологический базис теории» [там же]. Имеется в виду выбор единого прототипа, к которому можно было бы потом редуцировать все остальное типы бихевиоральных (поведенческих) систем. Позволю себе высказать предположение, что в качестве базиса выбора такого прототипа (фактически, языка, адекватного оптимизационному подходу или оптимизационному мышлению) как раз и может выступить предлагаемая концепция.

Реализуя указанный подход, необходимо, прежде всего, пересмотреть трактовку *адаптации* как процесса реагирования организма на изменение *только* окружающей его *внешней* среды (с целью улучшения его шансов на выживание и воспроизводство). Необходимо расширить это понятие, включив в него фактор реакции на изменения и *внутренней* среды организма (а также подобных ему целостных системных образований – клетки, биогеоценоза, Биогеосферы в целом и др.) – изменений, реализующих поиск наиболее энергетически предпочтительного его состояния.

Таким образом, предлагаемый подход к моделированию природных (замечу кстати – и общественных!) систем «достатично высокой сложности» базируется на единственной первичной предпосылке, не вытекающей ни из каких-либо «еще более первичных» оснований и/или теорий. Она состоит в предложении постулировать, что «первичное фундаментальное свойство природной системы в целом и всех её основных подсистем в отдельности – обобщенная адаптивность, т.е. свойство перманентной приспособляемости систем Природы и Человечества, на всех характерных уровнях их интеграции, к изменениям их внутренней и внешней среды». Обычно адаптивность рассматривают как свойство какой-либо системы приспосабли-

ваться лишь к внешним воздействиям на нее. Но в рамках предлагаемого подхода именно её расширенная, обобщенная трактовка, учитывающая изменения и внутренней среды, имеет кардинальное значение.

Как соотносятся «обычные» и оптимизационные модели

Может возникнуть вопрос: насколько оптимизационная модель соответствующего объекта отличается от типовой, знакомой по аналогичным приложениям? Не отменяет ли её появление имеющихся наработок? Проведем их сравнение (см. рис. 1).

Простейшую схему *типовой* модели некоторого *неиерархического* объекта можно представить себе следующим образом: $\mathbf{Y} = F(\mathbf{X})$ (обозначения приведены на рис. 1a).

Принципиальное отличие *оптимизационной* модели от типовой заключается во введении в её состав дополнительно всего 2-х основных блоков: блока вычисления целевой функции (ЦФ) энергетического характера $q(\mathfrak{a})$ и собственно блока оптимизации – генератора поисковой переменной \mathfrak{a} . При этом два последних блока *непосредственно* соединяются между собой: выход первого (целевая функция q) поступает на вход второго (оптимизатора), управляя им. Что же касается связи между выходом оптимизатора и входом блока вычисления целевой функции, то как раз здесь и находится все разнообразие элементов модели и связей между ними, характерное в том числе и для «типовой» модели объекта.

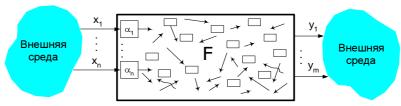
Таким образом, два вышеуказанных блока образуют некоторый «каркас» модели, обеспечивающий её приспособительное поведение. Все же остальное разнообразие функциональных связей, отражающих специфику конкретного объекта, введение такого «каркаса», вообще говоря, непосредственно не затрагивает. Конечно, рассмотрение объекта как самооптимизирующегося само по себе является источником новых идей относительно структуры его модели и предложений по наиболее адекватному их воплощению, но, повторюсь, оптимизационная модель – это соответствующее расширение (надстройка) типовой.

Другой ракурс их сравнения: переходная функция F типовой модели ϕ иксирована (именно её часто и называют моделью объекта). В оптимизационной же модели переходная функция $F_{\alpha(t)}$ зависит от поисковой переменной α , перманентно изменяющейся блоком оптимизации. В ситуации найденного устойчивого состояния объекта, когда поисковый процесс представляет собой лишь небольшие рыскания в районе экстремума оптимизируемой функции q, изменения переходной функции $F_{\alpha(t)}$ также будут весьма малыми. Тем самым воспроизводится вариант, близкий к типовому (т.е. с фиксированной переходной функцией F). В ситуации же *поиска* такого устойчивого состояния переходная функция F изменяется слишком значительно, и её фиксация невозможна из-за неминуемой потери принципиально важной информации о поведении моделируемого объекта.

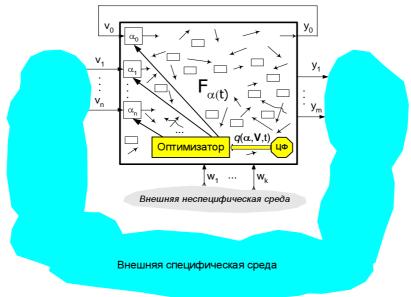
Коротко о механизме перархической адаптивной поисковой оптимизации живого

В весьма упрощенном виде схема типового элемента механизма иерархической адаптивной поисковой оптимизации живого – иерархический, или супра-контур – показан на рис. 2 на примере контура «МНОГО-КЛЕТОЧНЫЙ ОРГАНИЗМ-СЛОЖНЫЕ КЛЕТКИ». Совокупность ряда вложенных друг в друга подобных иерархических контуров и образуют указанный механизм. Последний является обобщением известного в технической кибернетике «неиерархического» механизма [8].

Для его описания необходимо ввести понятия основного уровня интеграции (ОУИ) в иерархии соответствующей системы, и промежуточного уровня интеграции (ПУИ). Принципиальные различия объектов, относящихся к ОУИ и ПУИ, состоят в том, что первые: а) существенно более целостны; б) оперируют соответствующей информацией — сигналами специфической функциональной активности: генерируют (как интегральный фактор, отражающий их специфическое поведение), передают (себе подобным, т.е. объектам того же яруса), воспринимают (от себе подобных), — а также и запоминают, но последнее делают и ПУИ; в) обладают фундаментальным свойством задавать целевые критерии поисковой оптимизации своего приспособительного поведения для "нижележащих" (т.е. непосредственно "вложенных" в ОУИ) элементов трех ярусов: двух смежных ПУИ и смежного нижележащего ОУИ.



а) простейшая схема *типовой* модели биологического объекта **Y**=F(**X**) Примечания: x_i - входы, y_j - выходы; α_i - коэффициенты чувствительности к входным воздействиям; отдельные стрелки и прямоугольники обозначают (здесь и ниже) произвольную внутреннюю структуру модели



б) упрощенная схема *оптимизационной* модели сложного биообъекта $\mathbf{Y} = \mathbf{F}_{\alpha(t)}(\mathbf{V}, \mathbf{W})$ Примечания: \mathbf{v}_i - специфические входы, \mathbf{w}_k - неспецифические входы, \mathbf{y}_j - выходы; α_i - коэффициенты чувствительности к специфическим входным воздействиям (задаются оптимизатором); $\mathbf{q}(\alpha_n\mathbf{v}_n,\alpha_1\mathbf{v}_1,\dots,\alpha_n\mathbf{v}_n,t)$ - целевая функция;

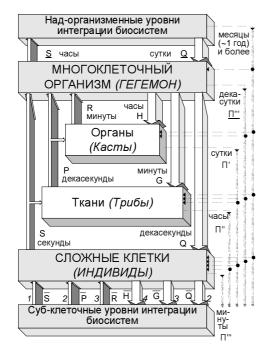
Рис.1. Упрощенные схемы типовой и оптимизационной моделей биообъекта.

Примеры систем ОУИ: Биогеосфера Земли, биогеоценозы, многоклеточные организмы, сложные клеткиэвкариоты (вслед за Ю.И. Чайковским [9] использую именно этот термин, а не «эукариоты»), прокариоты и т.п. Примеры систем ПУИ: биогеосферные компартменты (природные зоны), биогеосферные субкомпартменты (биомы), биогеоценотические компартменты (парцеллы), биогеоценотические субкомпартменты (популяции), компартменты многоклеточных организмов (органы/системы органов), субкомпартменты многоклеточных организмов (ткани), клеточные компартменты, клеточные субкомпартменты и т.п.

Иерархический, или супраконтур, содержит следующие компоненты и связи:

- два яруса ОУИ-систем, "высший", на котором единственный компонент (далее *ГЕГЕМОН*) задает критерии оптимизации для всего супраконтура, и "низший", на котором множество компонентов (далее *ИНДИВИДОВ*) реализуют поисковую активность в супраконтуре;
- две яруса ПУИ-систем, также реализующих поисковую активность, но интегрально: они представляют собой объединения (различной степени интеграции) множеств "низших" ОУИ-систем; причем ярус "высшей" из такой пары представляет собой сравнительно неоднородную структуру (далее *Касты*), а ярус "низшей" сравнительно однородную (далее *Трибы*);

три сугубо внутренние (для супраконтура) переменные типа "многие к одному", отражающие поисковую активность приспособительного поведения в рамках супраконтура, задаваемые компонентами обоих ПУИ-ярусов (Кастами, Трибами) и "низшим" ОУИ-ярусом (ИНДИВИДАМИ), и действующие соответственно на вышележащие компоненты обоих ПУИ-ярусов (Трибы, Касты) и "высшего" ОУИ-яруса (ГЕГЕМОН), компоненты верхнего в паре ПУИ-яруса (Касты) и "высшего" ОУИ-яруса (ГЕГЕМОН), и на компонент "высшего" ОУИ-яруса (ГЕГЕМОН);



Примечания: а) стрепки, направленные вверх, имеют структуру (отражают отношение) «многие - к одному», направленные вниз - «один - ко многим»; в) S,P,R - активности, П - память; в) 1 - сотни микросекунд, 2 - сантискунды, 3 - децискунды, 4 - секунды

Рис. 2. Упрощенная схема супраконтура иерархической адаптивной поисковой оптимизации «МНОГОКЛЕТОЧНЫЙ ОРГАНИЗМ-СЛОЖНЫЕ КЛЕТКИ».

- три сугубо внутренние (для супраконтура) переменные типа «один ко многим», отражающие целевые критерии приспособительного поведения в рамках супраконтура, задаваемые компонентом "высшего" ОУИ-яруса (ГЕГЕМОНОМ) и действующие на нижележащие компоненты ПУИ-ярусов (Касты, Трибы) и "низшего" ОУИ-яруса (ИНДИВИДЫ);
- три переменные типа "один ко многим", генерируемые внутри супраконтура, но влияющие не только на его компоненты, но и на компоненты всех супраконтуров, вложенных в рассматриваемый; эти переменные отражают системную память о траектории процесса приспособительного поведения компонентов каждого из ярусов рассматриваемого супраконтура, а также о параметрах собственно механизма поисковой оптимизации (лимитирования и закона изменения величины поискового шага, глубины памяти поискового алгоритма и т.п.).

В связи с тем, что любой ОУИ в таких иерархических системах выступает в двух ипостасях: как синоним «объекта» ($\Gamma E\Gamma EMOHA$) по отношению к включенным в него «низшим» ПУИ и ОУИ, и как синоним «элементарной поисковой единицы» (IHДИВИДA) по отношению к «высшим» ПУИ и ОУИ, в состав которых он сам входит, необходимо в каждом случае различать данные структурно-функциональные проявления. Для этого и введены термины $\Gamma E\Gamma EMOH$ (для акцентирования первой ипостаси объекта ОУИ) и IHДИВИ-ДЫ (для акцентирования второй). Тогда контур супраоптимизации (или супраконтур) определяется как совокупность четырех последовательно смежных уровней интеграции (ОУИ-ПУИ-ПУИ-ОУИ), наивысший из которых, или $\Gamma E\Gamma EMOH$, задает целевые критерии (CV и CV и CV

Отсюда естественным образом вытекает и понятие *супрасистемы* как системы, состоящей из совокупности (иерархии) *«супраконтуров»*, реализующих процесс *супраоптимизации*. Формальное описание процесса *иерархической адаптивной поисковой оптимизации* применительно к *«универсальному»* супраконтуру приведено в [6].

К главным особенностям предлагаемой задачи можно отнести следующие:

- 1. её принципиальную *технонтурность*. Каждый из трех целевых критериев замыкается на свою переменную (изменяющуюся в своем характерном темпе), переменные же находятся в иерархической связи между собой (реализующие их ярусы «вложены» один в другой). Очевидно, что отказ от объединения поисковых переменных \mathbf{S} в агрегаты \mathbf{P} и \mathbf{R} (плюс игнорирование системной памяти высших уровней в иерархии) сводит задачу (2) к задаче (1);
- 2. её принципиальную *незамкнутость*. Параметры системной памяти поискового алгоритма задаются вышележащими в иерархии системами различных уровней/ярусов, и правильность (более того и сама возможность!) решения задачи адаптации (средствами иерархической оптимизации) прямо зависит от их учета либо игнорирования. Конечно, с ростом номера иерархического уровня (считая его «снизу» «вверх» рис. 3) характерные времена происходящих там адаптивных процессов всякий раз существенно (примерно на порядок) увеличиваются. Это позволяет при их моделировании в строго ограниченных пределах заменять указанные переменные величины константами. Но тогда следует внимательно следить, чтобы зона прогнозирования поведения моделируемой иерархической системы не выходила бы за рамки, задаваемые подобными упрощениями;
- 3. её *структурную иерархичность*. Если говорить в терминах конкретной моделируемой системы, системная память память структур о прошлом *приспособительном* поведении –проявляется в форме *относительного постоянства* структуры объекта на протяжении *определенного* времени. По его истечении значение системной памяти сменяется другим, отражающим уже новый опыт приспособительного поведения элемента объекта (рассматриваемого яруса). То есть следует говорить не об изменении структуры объекта *вообще*, а конкретнее об изменении его структуры на рассматриваемом иерархическом уровне. Поскольку для различных ярусов иерархического объекта эти структуры выделяются по-иному и темпы системной памяти для них также различны.

Таким образом, приведенная выше задача иерархической поисковой оптимизации системы «достаточно высокой» сложности отражает фундаментальные для нее особенности «многотемповости» процессов её приспособительного поведения. Этот факт сам по себе, безотносительно к трудностям воспроизведения собственно структуры и связей моделируемой системы (по отношению к которым структура оптимизатора выступает в качестве своеобразного «каркаса»), чрезвычайно затрудняет (если не делает на современном этапе невозможными!) аналитические исследования поведения такой оптимизационной системы. Даже имитационные эксперименты с супрасистемами из нескольких супраконтуров представляются весьма сложной проблемой. В частности, визуализацию их результатов – в реальном модельном времени – весьма трудно реализовать. Это показали исследования «алгоритмической» модели живой клетки, разработанной на данной основе, т.е. включающей механизм поисковой оптимизации (конкретно – матричного случайного поиска) энергетического критерия её функционирования [10].

Очевидно, что *непосредственное* модельное воспроизведение *совокупности* всех этих процессов в природной (живой) либо в социально-технологической системах современными средствами вычислительной техники невозможно. Даже в далекой перспективе, если и когда будут созданы иерархически организованные параллельные вычислительные системы и сети, можно надеяться на появление возможности моделировать лишь те или иные иерархические фрагменты таких систем. Но и в ожидании воплощения этих перспектив вполне возможно и целесообразно использовать *идеологию* предлагаемой «оптимизационной» концепции в качестве языка системного описания как собственно функционирования систем «достаточно высокой» сложности, так и их «метаэволюционного» развития.

О теориях биологической эволюции

Ситуация с «общепринятостью» эволюционных теорий представляется на сегодняшний день парадоксальной. С одной стороны, дарвинизм и его сегодняшняя развитая форма – синтетическая теория эволюции (СТЭ) – подвергаются серьезной критике в весьма важных аспектах этих теорий. С другой стороны – реакция биологов-эволюционистов на эту критику практически отсутствует: в научных изданиях эта критика игнорируется, в учебниках (и не только в школьных, но и в вузовских) о дарвинизме пишут как о полностью подтвержденной фактами научной теории, и т.д., и т.п.

Между тем, вот что, например, пипут о трудностях СТЭ и неодарвинизма. Чтобы не быть голословным, приведу лишь две цитаты, в которых расставлены большинство нужных, с моей точки зрения, акцентов: «Существенный вклад в формирование системологии внес Чарльз Дарвин, обосновав принцип естественного отбора и дивергенции в эволюции. Более чем вековая дискуссия о значении естественного отбора в эволюции, была завершена признанием этого принципа как основополагающего в области саморазвития материи, в

том числе и на молекулярном уровне. Но в рамках дарвиновской теории не была решена проблема появления новых признаков. Эта наиболее сложный вопрос биологии стал ареной борьбы между материалистическими и идеалистическими направлениями в естествознании. Пожалуй, первым исследователем, постулировавшем эволюцию как закономерное свойство самих систем к саморазвитию, был Л.С.Берг (1922). Он полагал, в противовес господствующим представлениям дарвинистов, что "...создание все более и более совершенных форм есть имманентное свойство живой природы", что основой является "...внутреннее начало, лежащее в самих организмах, а не привносимое путем соединения частей и воздействий внешнего мира". Эти представления до сих пор не получили широкого признания. Согласно современной СТЭ, в основе её лежит накопление случайно возникших под влиянием повреждающих факторов среды повреждений наследственного аппарата. Но такой механизм не универсален (Воронцов, 2000) и тем более он не может быть распространен на предбиологические этапы эволюции, что сужает рамки эволюции организменным уровнем организации материи» [11].

Еще цитата (из послесловия к новому изданию, 2002 года, «Суммы технологии» Станислава Лема): «Минувшие со времени создания "Суммы..." (1967 год – С. Г.) десятилетия никак не отразились на общих представлениях о процессе эволюции, в то время как история жизни на Земле была в значительной степени переписана... на сегодняшний день не существует сколько-нибудь разумной рабочей гипотезы, позволяющей объяснить биогенез и "запуск" механизма биологической эволюции (...) Критику "классической теории эволюции" можно продолжать и далее. Следует, однако, сказать прямо, что альтернативного механизма пока не предложено. Применение к эволюционному процессу аппарата теории систем позволило получить ряд любопытных результатов, но все они носят "гомеостатический" (курсив мой – С. Г.) характер, то есть описывают устойчивость, а не изменчивость экосистем» ([12], стр. 658-661).

Надежду на скорый прогресс в этом вопросе вселяет тот факт, что понятие «поисковый оптимизатор» само по себе является кардинальным обобщением довольно распространенного понятия «гомеостат», позволяя достигать не только устойчивости в системе, но и её изменчивости (посредством перманентного переключения знака «обратной связи» в соответствующие моменты времени). А реализация «поискового оптимизатора» с памятью, т.е. в форме адаптивного случайного поиска, тем более в иерархическом варианте, позволяет надеяться на получение при моделировании эволюционных механизмов – с его помощью – принципиально новых результатов.

Таким образом, с моей точки зрения, путь к преодолению этих проблем находится именно в русле системного подхода, а конкретно — в синтезе дарвинизма и номогенеза Л. С. Берга на информатико-кибернетической основе. То есть на базе представления об эволюции как о процессе поисковой оптимизации живого по энергетическим критериям, которая необходимо учитывает системную память на всех уровнях иерархии живого [6]. Другими словами, указанный синтез может быть произведен на базе концепции иерархической адаптивной поисковой оптимизации природных систем. Последняя позволяет снять многолетнее недоразумение — рассмотрение дарвинистских и номогенетических теорий эволюции живого как альтернативных, объединяя центральные положения указанных подходов в рамках единой схемы. Именно объединение данных теорий эволюции в общую теорию на базе кибернетических (оптимизационных) представлений дает возможность с единых позиций объяснить многочисленные эмпирические факты, часть из которых противоречат дарвинизму, а часть — номогенезу.

Помимо прочего, этот синтез позволяет при объяснении процессов *эволюции* живого отказаться от искусственного, неоперационального и тавтологичного понятия «приспособленности» (кстати говоря, формулировку которого Дарвину предложил Спенсер – см. [13], стр. 73), заменив его вполне измеримым и универсальным понятием *энергетических характеристик* соответствующих биологических систем.

Кроме того, иерархическое «оптимизационное» представление о приспособительном поведении биологических систем позволяет не только увязать в одно целое существенно «разновременные» процессы эволюции и «классической» адаптации как относящиеся к различным уровням интеграции живого (т.е. дав им общую кибернетическую трактовку в рамках данной схемы). «Направленность» и «канализируемость» эволюционного процесса трактуются здесь соответственно как тенденция стремления (в среднем) к экстремумам целевых функций (энергетического характера) на всех уровнях интеграции живого, и как влияние целевых ограничений на этот процесс. Наличие же системной памяти на всех ярусах иерархии реализует операцию кумулятивного накопления опыта такого эволюционного процесса.

Одним из следствий такого представления является уточнение используемых понятий: термин «эволюция» отличается от термина «адаптация организма» лишь значениями характерных времен и пространственных размеров: «эволюция» относится к иерархическим контурам «БИОГЕОЦЕНОЗ-ОРГАНИЗМЫ» и «БИО-

ГЕОСФЕРА-БИОГЕОЦЕНОЗЫ», а «адаптация организма» – естественно, к иерархическому контуру «ОР-ГАНИЗМ-КЛЕТКИ». *Кибернетически* же эти процессы *принципиально*, т.е. в своей *основ*е, не отличаются.

Другим таким следствием является необходимость различать в общепринятом понятии «эволюция» два совершенно различных аспекта: эволюцию *ужее существующей* иерархической структуры, и возникновение *новых* уровней в иерархии живого (в историческом прошлом Земли, и, предположительно, в будущем). Этот последний процесс удобно назвать «*метаэволюцией*» живого [6]. Сравнительный анализ процессов метаэволюции живого, неживого и «социально-технологического» представлен в [14].

Некоторые следствия предлагаемого подхода

Итак, изложенные выше факты свидетельствуют в пользу утверждения, что схема *иерархического механизма адаптивной поисковой оптимизации* как кибернетическая система обладает своим собственным набором характеристик (отражающих особенности его структуры и поведения безотносительно к субстрату его материальной реализации), и этот факт является мощным источником сведений для структуризации соответствующих систем «достаточно высокой сложности» (не только биологических, но и социальнотехнологических, экономических и т.п.), выявлению в них иерархий пространственных, временных и количественных, единой трактовке в кибернетических терминах места и роли переменных, зачастую представляющихся весьма и весьма различными (опять-таки из-за несовпадения субстрата и пространственно-временных характеристик) и других аналогий.

Кратко приведу некоторые следствия предлагаемой концепции **методологического** и, в какой-то степени, **философского характера** [6]:

- 1. Системность теперь не может лишь декларироваться, а при этом все усилия направляться на выявление тех или иных узколокальных свойств природных объектов. Volens nolens, «узким» специалистам не следует забывать о том, что трудности или даже «тупики» в исследованиях могут иметь объективный характер как результат неучета в экспериментах либо при построении локальных моделей относительно сложного явления именно системных взаимодействий (оптимизационного характера) в иерархии живого.
- 2. Системное целеполагание (зачастую обозначаемое терминами «телеология» или «телеономия») никоим образом не означает привлечения для объяснения системных явлений каких бы то ни было сверхъестественных причин и/или субъектов. В рамках естественнонаучного знания достаточно понятий прежде всего, понятия энергии, привлечение которых позволяет описывать целенаправленные процессы вполне адекватно. Более того, необходимо уйти от превалирующего на сегодня представления, что «для многих биологических систем понятие цели управления не определено (напр., что является целью эволюции?). Поэтому зачастую задание критерия или цели управления является лишь удобным приемом, позволяющим построить замкнутую модель при недостатке конкретной информации» ([15], C.254). Подобную трактовку следует пересмотреть и рассматривать процедуру задания цели управления и реализации её алгоритма как механизм, имманентно присущий соответствующему биообъекту (относящемуся к основному уровню биологической интеграции).
- 3. Система живой природы *не* пассивна, а проявляет *активность* в достижении собственных целей, и любые возмущающие воздействия на них приводят лишь к сопротивлению (как может быть интерпретирован принцип Ле-Шателье). Именно это позволяет объяснить феномен так называемого «антиинтуитивного» [16] поведения систем «достаточно высокой» сложности. С позиций предлагаемой концепции причина подобного поведения активное противодействие каждой такой системы любым воздействиям, «выталкивающим» её из экстремального наиболее предпочтительного с её точки зрения состояния. В условиях отсутствия у исследователя модели целевого критерия такой системы и алгоритма его достижения, предсказать её будущую траекторию весьма затруднительно, если не невозможно.
- 4. Важно различать *адаптивность* как свойство приспособляемости некоторого биообъекта к изменениям *только* его внешней среды, и *обобщенную адаптивность* как свойство перманентной приспособляемости систем природы и Человечества (на всех характерных уровнях их интеграции) к изменениям их как внешней, так и внутренней сред.
- 5. *Принцип причинности* в привычном понимании выполняется лишь в ограниченных (в пространстве и во времени) пределах, а при выходе за такие пределы *непосредственная* связь «причин» и «следствий» начинает все более и более нарушаться: ибо что есть «причина» и что есть «следствие» в *контуре*, тем более даже не «замкнутом», а развивающемся «по спирали»? И при этом с существенным запаздыванием процессов в нем?
- 6. **Использование** при изучении систем «достаточно высокой» сложности как целого большинства привычных математических представлений (в частности, в виде описания исследуемых процессов с помо-

щью систем дифференциальных уравнений, как обыкновенных, так и в частных производных) весьма ограничено. Конечно, их использование возможно, но при изучении лишь отдельных фрагментов «разрезанного» контура, с искусственно фиксированными в процессе анализа «входами» и «выходами», т.е. «причинами» и «следствиями».

7. Случайность, имманентно присущая механизму иерархической адаптивной поисковой оптимизации живого, возникает естественным образом как проявление иерархичности самой системы живого. Это можно интерпретировать как обобщение достаточно часто используемой гносеологической трактовки случайности как меры нашего незнания об объекте. Различие состоит в том, что в иерархической оптимизационной подсистеме некоторого уровня интеграции I даже при вполне регулярном поведении каждой из составляющих её подсистем $i_1,...i_n$ (обладающих свойством активности) рассмотрение такого поведения «в целом» – одновременно и всех сразу, т.е. «с точки зрения» подсистемы именно высшего уровня I, — может выглядеть только как случайное: у подсистемы I просто нет так называемого «планшета» для фиксации поведения всех её подсистем $i_1,...i_n$.

Список литературы

- 1. Турчин В. Ф. Феномен науки. Кибернетический подход к эволюции. М.: ЭТС, 2000. 368 с.
- 2. Седов А. Е. Метафоры в генетике // Вестник Российской Академии наук, 2000, том 70, № 6, С. 526-534.
- 3. Моисеев Н. Н. Быть или не быть... Человечеству? М.: 1999. 289 с.
- 4. Heylighen F., Joslyn C. Cybernetics and Second-Order Cybernetics // in: R.A. Meyers (ed.), Encyclopedia of Physical Science & Technology, Vol. 4 (3rd ed.), Academic Press, New York, 2001. pp. 155-170. http://pespmc1.vub.ac.be/Papers/Cybernetics-EPST.pdf
- 5. Растригин Л. А. Системы экстремального управления. М.: Наука, 1974. 632
- 6. Гринченко С. Н. Системная память живого (как основа его метаэволюции и периодической структуры). М.: ИПИРАН, Мир, 2004, 512 с. см. также http://urss.ru
- 7. Разумовский О. С. Концепция оптимологии. Новосибирск: ЦСА, 1998. 64 с.
- 8. Растригин Л. А. Адаптация сложных систем. Методы и приложения. Рига: Зинатне, 1981. 375 с. см. также http://dssg.cs.rtu.lv/ru/index.html
- 9. Чайковский Ю. В. Эволюция. Часть 7. Загадка начала жизни. ("Биология" комплект изданий "Первое сентября". 1999, № 11.) http://evolution.atheism.ru/library/beginninglife.html
- 10. Гринченко С. Н., Загускин С. Л. Механизмы живой клетки: алгоритмическая модель. М.: Наука, 1989. 232 с.
- 11. Краснощеков Г. П., Розенберг Г. С. Экология в цитатах и афоризмах. Тольятти: 2001 http://www.tolcom.ru/kiril/Rozenberg/Ea1.htm
- 12. Переслегин С. Того, что достаточно для Геродота, мало для Герострата // Послесловие редактора к изданию: Лем С. Сумма технологии. М.: Изд-во АСТ, СПб: Terra Fantastica, 2002, 670 с.; С.643-668.
- 13. Дарвин Ч. Происхождение видов путем естественного отбора. М.: Тайдекс Ко, 2003, 496 с. см. также http://charles-darwin.narod.ru/origin-content.html
- 14. Гринченко С. Н. Является ли метаэволюция Вселенной запрограммированным и целенаправленным процессом? // Электронный журнал «Исследовано в России», 17, стр. 164-195, 2005 г., http://zhurnal.ape.relarn.ru/articles/2005/017.pdf
- 15. Биология. Большой энциклопедический словарь. М.: Большая Российская энцикл., 1999. 864 с.
- 16. Forrester J. W. Counterintuitive Behavior of Social Systems // Technology Review, 1971, Vol. 73, No. 3, pp. 53-68. Also appears as Chapter 14, pages 211-244, in the author's Collected Papers 1975; and as Chapter 1, pp. 3-30, in Toward Global Equilibrium: Collected Papers, 1973, Dennis L. Meadows, ed., both from Waltham, MA. Pegasus Communications (русский перевод: Форрестер Дж. В. Антиинтуитивное поведение сложных систем // Современные проблемы кибернетики. М.: Знание, 1977, С. 9-25).

Отличимость геометрических лабиринтов

Грунская В. И.

В настоящее время актуальны задачи, связанные с анализом различных сред с помощью блуждающих по ним агентов. В таких задачах рассматриваются различные модели сред. Одной из них являются графы, которые можно рассматривать как топологическую модель среды [1]. В зависимости от решаемых задач рассматриваются как обычные графы, так и графы с отмеченными дугами и вершинами. Особое место среди графовых моделей среды занимают п-мерные лабиринты [2], дуги которых параллельны единичным векторам п-мерного евклидова пространства. Они интересны еще и тем, что являются частным случаем автоматных сред. В работе [3] для 2-мерных лабиринтов решались задачи, аналогичные классическим задачам теории автоматов: отличимости лабиринтов и их вершин. Выло показано, что оценка длины различающего слова для вершин лабиринта с т вершинами равна $\left[\frac{m}{2}\right]$, что меньше как результата Мура для конечных автоматов, так и результатов, полученных для неориентированных графов [4,5].

В настоящей работе исследуются классы 2-мерных лабиринтов, дуги которых параллельны векторам некоторого произвольно выбранного множества. Показано, что оценка длины различающего слова для лабиринтов из таких классов не зависит от количества выбранных векторов и их направления, а определяется "правильностью" разметки дуг лабиринта. Для правильно размеченных лабиринтов (d-лабиринтов) она совпадает с оценкой, полученной для шахматных лабиринтов. Кроме того, показано, что d-лабиринты являются приведенными, и отношения эквивалентности и изоморфизма для них совпадают.

Рассмотрим двумерное евклидово пространство с декартовой системой координат и целочисленную решетку \mathbf{Z}^2 в нем. Элементы (x,y) множества \mathbf{Z}^2 будем обозначать v. Будем считать, что в пространстве задано множество попарно не коллинеарных направляющих векторов $\mathbf{O} = \{\mathbf{o}_1,...,\mathbf{o}_q\}$

Пусть $B = \{b_1, ..., b_q, b_1^{-1}, ..., b_q^{-1}\}$ - некоторый алфавит, и каждому вектору **o** из **O** сопоставлен единственный символ b из B, который называется его направлением и направлением всех векторов, сонаправленных вектору **o**. Символ b^{-1} будем считать направлением любого вектора, противоположно направленного вектору **o**. Символы b^{-1} и b назовем противоположными. Обозначим через θ пустой символ, $\theta \notin B$, $B^0 = B \cup \{\theta\}$, $A = 2^B \setminus \emptyset$.

Отмеченным графом G=(V,E,a,b) назовем связный ориентированный конечный псевдограф, у которого V - множество вершин, E - множество дуг, $a:v\in V\to a\in A$ - функция разметки вершин, $b:(v,v')\in E\to b\in B$ - функция разметки дуг. Через |G| обозначим число вершин графа G, через A_G и B_G - множества отметок всех вершин и дуг графа G соответственно.

s-лабиринтом L=(V,E,a,b) назовем симметричный планарный отмеченный граф без петель и кратных дуг, обладающий следующими свойствами: отметки b(v,u) всех дуг, исходящих из вершины v, попарно различны; отметка a(v) каждой вершины v есть множество отметок всех исходящих из нее дуг; отметки дуг (v,u) и (u,v) противоположны.

Отмеченный граф G=(V,E,a,b) назовем геометрическим, если его множество вершин V есть подмножество \mathbf{Z}^2 , дуги параллельны направляющим векторам из множества \mathbf{O} , любая пара дуг, не являющихся противоположными, может иметь не больше одной общей точки, причем если она есть, то является началом или концом каждой дуги.

Геометрический граф назовем d-графом, если отметка каждого каждой его дуги соответствует ее направлению.

g-лабиринтом (d-лабиринтом) назовем s-лабиринт, являющийся геометрическим графом (d-графом).

Слово $t=(a_1,b_1)\dots(a_m,b_m)$ в алфавите $A\times B^0$ назовем правильным, если для всех $i\in\{1,\dots,m-1\}$ выполнено $b_i\in a_i,\,b_i^{-1}\in a_{i+1},$ и $b_m=\theta$. В дальнейшем будем рассматривать только правильные слова.

Маршрутом p назовем последовательность вершин и дуг $p = v_1(v_1, v_2)v_2 \dots v_m(v_m, v_{m+1})v_{m+1}$. Путем назовем маршрут, в котором все дуги попарно различны, простым путем - путь, в котором все вершины попарно различны. Будем говорить, что слово $t = (a_1, b_1) \dots (a_{m+1}, b_{m+1})$ и маршрут p в отмеченном графе G соответствуют друг другу, если для всех $i \in \{1, \dots, m+1\}$ выполнено $a_i = a(v_i)$ и для всех $i \in \{1, \dots, m,\}$ выполнено $b_i = b(v_i, v_{i+1})$.

Каждой вершине v отмеченного графа G поставим в соответствие множество слов λ_v , соответствующих всевозможным маршрутам, начинающимся в вершине v.

Отмеченный граф G будем называть детерминированным, если для любой его вершины v каждому слову t из λ_v соответствует единственный маршрут с началом в вершине v. Очевидно, что любой s-лабиринт является детерминированным.

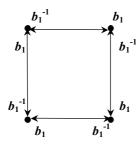


Рис. 1: g-лабиринт, все вершины которого неотличимы

Вершины v отмеченного графа G и v' отмеченного графа G' назовем неотличимыми, если множества λ_v и $\lambda_{v'}$ совпадают. В противном случае вершины v и v' назовем отличимыми, а слово $t \in \lambda_v \oplus \lambda_{v'}$ - различающим. Очевидно, что отношение неотличимости есть отношениями эквивалентности.

Отмеченный граф G назовем приведенным, если любая пара его вершин отличима.

Отмеченные графы G и G' назовем эквивалентными, если для любой вершины графа G найдется неотличимая от нее вершина графа G' и для любой вершины графа G' найдется неотличимая от нее вершина графа G.

Отмеченные графы G = (V, E, a, b) и G' = (V', E', a', b') назовем изоморфными, если существует такое взаимно однозначное соответствие $\varphi: V \to V'$, что для любых вершин $v, u \in V$ дуга (v, u)принадлежит E тогда и только тогда, когда $(\varphi(v), \varphi(u)) \in E'$, и $a(v) = a'(\varphi(v)), a(u) = a'(\varphi(u)),$ $b(v, u) = b'(\varphi(v), \varphi(u)).$

Пусть L - некоторый d-лабиринт. Вершины лабиринта L с отметкой $\{b_1,\dots,b_q,b_1^{-1},\dots,b_q^{-1}\}$ назовем внутренними. Вершину, в отметке которой отсутствует символ b_i $(b_i^{-1}), i=1,...,q$, назовем і-граничной ((-і)-граничной).

Пусть задана вершина v d-лабиринта L. Обозначим через P_v^i множество всех простых путей из v в i-граничные вершины d-лабиринта L, $P_v^b = (\bigcup_{i \in \{1,q\}} P_v^i) \cup (\bigcup_{i \in \{1,q\}} P_v^{-i})$, а через λ_v^b обозначим множество всех слов, соответствующих всем путям из P_v^b .

Теорема 1. Два приведенных отмеченных графа эквивалентны тогда и только тогда, когда они изоморфны.

Пусть L=(V,E,a,b) - некоторый s-лабиринт. Разобьем множество V на классы V_0,\dots,V_k попарно неотличимых вершин. Отметкой a' класса будем считать отметку входящих в него вершин. Будем говорить, что пара $(V_i,V_j)\in E'$ тогда и только тогда, когда для любой вершины $v_i \in V_i$ существует вершина $v_j \in V_j$ такая, что $(v_i, v_j) \in E$. Положим $b'(V_i, V_j) = b(v_i, v_j)$. Граф $L/_{\sim}=(\{V_0,\ldots,V_k\},E',a',b')$ назовем фактор-графом s-лабиринта L. Очевидно, что фактор-граф sлабиринта является приведенным отмеченным графом и в общем случае не является *s*-лабиринтом.

Теорема 2. Для любого s-лабиринта L существует единственный эквивалентный ему отмеченный граф с минимальным числом вершин, и этот минимальный граф изоморфен фактор-графу s-лабиринта L.

Теорема 3. Любой д-лабиринт является приведенным.

Пример на рис.1 показывает, что существуют не приведенные **g**-лабиринты.

Теорема 4. Для д-лабиринтов следующие утверждения эквивалентны.

- 1. Некоторые вершины v d-лабиринта L u v' d-лабиринта L' неотличимы.
- 2. Множества λ_v^b и $\lambda_{v'}^b$ равны.
- 3. d-лабиринты L и L' эквивалентны. d . d-лабиринты L и L' изоморфны.

Теорема 5. 1) Пусть L - некоторый g-лабиринт, $v, v' \in V$, $v \neq v'$. Вершины v u v' неотличимы тогда и только тогда, когда они неотличимы никаким словом длины $|L| - |A_L| + 1$.

2) Пусть L - некоторый d-лабиринт, $v, v' \in V$, $v \neq v'$. Вершины v и v' отличимы словом длины $\left[\frac{|L|}{2}\right]$.

Существуют примеры, показывающие, что обе оценки достижимы.

Список литературы

- [1.] Kuipers B. The Spatial Semantic Hierarchy // Artificial Intelligence, 2000, -v.119, n. 1–2, -pp. 191-233.
- [2.] Килибарда Г., Кудрявцев В. Б., Ушчумлич Щ. Независимые системы автоматов в лабиринтах // Дискретная математика, 2003, Т. 15, вып. 2, С. 3-39.
- [3.] Грунская В. И. Об отличимости плоских шахматных лабиринтов. // Интеллектуальные системы. -2004.- Т.8, С.457-464.
- [4.] Сапунов С. В. Эквивалентность помеченных графов // Труды ИПММ НАНУ, 2002, -т. 8, с. 162-167.
- [5.] Сапунов С. В. Контроль детерминированных графов // Труды ИПММ НАНУ, 2003, -т. 8, с. 106-110.
 - [6.] Харари Ф. Теория графов. М.: Мир, 1973 300с.
- [7.] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию конечных автоматов. М.: Наука, 1985. 320 с.

Многокритериальная комбинаторная задача разбиения в условиях неопределенности ¹

Е. Е. Гуревский, В. А. Емеличев,

Белорусский государственный университет, пр. Независимости, 4, Минск 220050, Беларусь E-mail: eugen_eugen@tut.by, emelichev@bsu.by

Рассматривается многокритериальный (векторный) вариант комбинаторной задачи разбиения, которая состоит в разделении конечного множества чисел на два подмножества так, чтобы суммы элементов в подмножествах отличались минимальным образом. Получены оценки снизу и сверху радиуса того типа устойчивости к возмущениям параметров векторного критерия в метрике l_{∞} , который является дискретным аналогом свойства полунепрерывности сверху по Хаусдорфу многозначного отображения, задающего паретовскую функцию выбора. Показано, что нижняя оценка достижима.

Ключевые слова: многокритериальная оптимизация, комбинаторная задача разбиения, устойчивость, радиус устойчивости.

1. Введение

При решении практических задач оптимизации приходится учитывать различные факторы неопределенности, такие как неточность входной информации, неадекватность математических моделей реальным процессам, ошибки округления, погрешность вычислений и т. п. В этой связи широкое распространение дискретных оптимизационных моделей в последние десятилетия стимулировало внимание многих специалистов к изучению разнообразных аспектов теории некорректных задач, и, в частности, к вопросам устойчивости. Устойчивость применительно к задачам оптимизации (как скалярным, так и векторным) – это существование такой окрестности начальных данных в пространстве параметров задачи, что по отношению к исходной всякая "возмущенная" задача с параметрами из этой окрестности обладает некоторым наперед заданным свойством инвариантности. Существуют различные типы устойчивости задач дискретной оптимизации (см., например, [1]–[6], а также аннотированную библиографию в [7]). Чаще всего под устойчивостью оптимизационной задачи понимается одно из свойств непрерывности или полунепрерывности (например, по Бержу или Хаусдорфу) многозначного отображения, которое набору параметров задачи ставит в соответствие множество искомых решений (см., например, [1], [4], [8], [9]). В случае дискретного множества решений такое определение устойчивости легко может быть переформулировано в терминах существования "шара устойчивости". Например, свойство полунепрерывности сверху по Хаусдорфу отображения, задающего принцип оптимальности дискретной задачи, эквивалентно свойству непоявления новых оптимальных решений задачи при "малых" возмущениях ее параметров. Это свойство и есть устойчивость задачи в узком смысле, именно о нем и будет идти речь ниже.

¹Работа поддержана Межвузовской программой Республики Беларусь "Фундаментальные и прикладные исследования" (грант 492/28).

Настоящая работа лежит в русле направления, связанного с конструктивным подходом к проблеме устойчивости, состоящим в получении количественных характеристик устойчивости. Одна из таких характеристик, называемая обычно радиусом устойчивости, определяется как предельный уровень упомянутых выше возмущений в метрическом пространстве параметров задачи. В качестве возмущающих параметров обычно выступают коэффициенты скалярного или векторного критерия. Как правило, результатом таких исследований являются формальные выражения, оценки или алгоритмы вычисления радиуса устойчивости (см., например, [2], [3], [5], [10]–[15]).

Продолжая цикл исследований, посвященный проблеме устойчивости векторных комбинаторных задач и отраженный в [16]–[22], рассмотрим многокритериальный вариант классической комбинаторной задачи разбиения. Эта задача состоит в следующем. Набор из нескольких чисел требуется разбить на два подмножества таким образом, чтобы суммы элементов в подмножествах отличались минимальным образом. В случае, когда элементы набора положительны, эта задача эквивалентна задаче теории расписаний, состоящей в распределении независимых работ по двум идентичным процессорам так, чтобы время, когда заканчивается последняя выполненная работа, было минимальным (см., например, [23], [24]). В теории расписаний эта задача обозначается $P|\cdot|C_{\rm max}$.

Целью работы является исследование предельного уровня независимых возмущений параметров векторного критерия задачи разбиения, неприводящих к появлению новых парето-оптимальных решений. Получены нижняя и верхняя оценки радиуса устойчивости и показано, что нижняя оценка является достижимой.

2. Основные определения и обозначения

Пусть $x=(x_1,x_2,\ldots,x_n)^T\in Q^n,\ Q=\{-1,1\},\ n\geq 2,\ C_i-i$ -ая строка матрицы $C=[c_{ij}]_{m\times n}\in \mathbf{R}^{m\times n},\ m\geq 1,\ i\in N_m=\{1,2,\ldots,m\}.$

Под m-критериальной (векторной) задачей разбиения $Z^m(C)$: $\min\{f(x,C): x\in Q^n\}$, где $f(x,C)=(|C_1x|,|C_2x|,\ldots,|C_mx|)$, будем понимать задачу поиска множества эффективных решений (множества Парето) $P^m(C)=\{x\in Q^n: \pi(x,C)=\emptyset\}$, где $\pi(x,C)=\{x'\in X: f(x,C)\geq f(x',C)\$ & $f(x,C)\neq f(x',C)\}$.

В силу конечности Q^n множество $P^m(C)$ непусто при любой матрице $C \in \mathbf{R}^{m \times n}$. Заметим, что при $x^0 \in P^m(C)$ решение $-x^0$ также является эффективным ввиду $f(x^0,C) = f(-x^0,C)$. Поэтому множество Парето задачи $Z^m(C)$ всегда имеет четное число решений.

Будем исследовать устойчивость множества $P^m(C)$ к возмущению параметров векторной функции f(x,C), под которым будем понимать произвольное независимое изменение матрицы C. Это изменение будем моделировать путем прибавления к матрице C возмущающих матриц. Для этого в пространстве \mathbf{R}^k произвольной размерности $k \in \mathbf{N}$ зададим две метрики l_1 и l_∞ , т. е. под нормами вектора $z = (z_1, z_2, \ldots, z_k)^T \in \mathbf{R}^k$ будем понимать соответственно числа

$$||z||_1 = \sum_{j \in N_k} |z_j|, \quad ||z||_{\infty} = \max_{j \in N_k} |z_j|,$$

а под нормой матрицы – норму вектора, составленного из ее элементов.

Для любого числа $\varepsilon > 0$ введем множество возмущающих матриц в пространстве $\mathbf{R}^{m \times n}$ с метрикой $l_{\infty} : \Omega(\varepsilon) = \{C' \in \mathbf{R}^{m \times n} : ||C'||_{\infty} < \varepsilon\}.$

Как обычно [1], [3], [4], [6], [17], [19], задачу $Z^m(C)$ назовем устойчивой (по функционалу), если

$$\Xi(C) = \{ \varepsilon > 0 : \forall \ C' \in \Omega(\varepsilon) \ (P^m(C + C') \subseteq P^m(C)) \} \neq \emptyset.$$

Тем самым устойчивость задачи является дискретным аналогом свойства полунепрерывности сверху по Хаусдорфу в точке $C \in \mathbf{R}^{m \times n}$ многозначного оптимального отображения $P^m : \mathbf{R}^{m \times n} \to 2^{Q^n}$, т. е. точечно-множественного отображения, которое каждому набору параметров задачи (матрице C) ставит в соответствие множество Парето $P^m(C)$.

Радиусом устойчивости задачи $Z^m(C)$ называется число

$$\rho^{m}(C) = \begin{cases} \sup \Xi(C), \text{ если } \Xi(C) \neq \emptyset, \\ 0, \text{ если } \Xi(C) = \emptyset. \end{cases}$$

Другими словами, радиус устойчивости – это предельный уровень возмущений элементов матрицы C в пространстве $\mathbf{R}^{m \times n}$ с метрикой l_{∞} , которые не приводят к появлению новых эффективных решений.

Очевидно, что при выполнении равенства $P^m(C) = Q^n$ радиус устойчивости задачи $Z^m(C)$ равен бесконечности. Задачу $Z^m(C)$, для которой множество $\overline{P}^m(C) := Q^n \backslash P^m(C)$ непусто, назовем нетривиальной.

Будем использовать обозначение

$$\operatorname{sg} t = \begin{cases} 1, & \operatorname{если} t \ge 0, \\ -1, & \operatorname{если} t < 0, \end{cases}$$

и импликацию

$$(\exists q \in Q \ \forall q' \in Q \ (qt > q't')) \Rightarrow |t| > |t'|, \tag{1}$$

которая с очевидностью выполняется для чисел $t, t' \in \mathbf{R}$.

3. Оценки радиуса устойчивости

Теорема. Пусть

$$\varphi^{m}(C) = \min_{x \in \overline{P}^{m}(C)} \max_{x' \in \pi(x,C)} \min_{i \in N_{m}} \min_{q \in Q} \frac{|C_{i}(x + qx')|}{||x + qx'||_{1}}.$$
 (2)

Тогда для радиуса устойчивости $\rho^m(C)$ нетривиальной задачи $Z^m(C), m \geq 1$, справедливы оценки

$$\varphi^m(C) \le \rho^m(C) \le \min\{||C_i||_{\infty} : i \in N_m\},\$$

причем $\rho^m(C) = \varphi^m(C)$, если $|P^m(C)| = 2$.

Доказательство. Легко видеть, что $\varphi := \varphi^m(C) \ge 0$.

Сначала докажем неравенство $\rho^m(C) \geq \varphi$. Будем предполагать, что $\varphi > 0$ (в противном случае неравенство $\rho^m(C) \ge \varphi$ очевидно). Пусть $C' \in \Omega(\varphi)$. Тогда в соответствии с определением (2) числа φ для всякого решения $x \in \overline{P}^m(C)$ существует такое решение $x^* \in \pi(x,C)$, что для любых $i \in N_m$ и $q \in Q$

$$||C'||_{\infty} < \varphi \le \alpha_i(x, x^*, q), \tag{3}$$

где

$$\alpha_i(x, x^*, q) = \frac{|C_i(x + qx^*)|}{||x + qx^*||_1}.$$

Это значит, что

$$|C_i(x+qx^*)| > 0,$$

и поэтому ввиду $x^* \in \pi(x, C)$ получаем

$$|C_i x| > |C_i x^*|.$$

Отсюда, полагая $\sigma_i = \text{sg } C_i x$, легко убеждаемся, что

$$C_i(\sigma_i x + qx^*) = |C_i(x + \sigma_i qx^*)|.$$

Тогда ввиду (3) имеем

$$\sigma_i(C_i + C_i')x + q(C_i + C_i')x^* = C_i(\sigma_i x + qx^*) + \sigma_i C_i'(x + \sigma_i qx^*) \ge |C_i(x + \sigma_i qx^*)| - ||C_i'||_{\infty} \cdot ||x + \sigma_i qx^*||_1 > |C_i(x + \sigma_i qx^*)| - \alpha_i(x, x^*, \sigma_i q)||x + \sigma_i qx^*||_1 = 0.$$

Из доказанного следует $\sigma_i(C_i + C_i')x > q(C_i + C_i')x^*, i \in N_m, q \in Q$. Откуда, вследствие указанной выше импликации (1), выводим

$$|(C_i + C_i')x| > |(C_i + C_i')x^*|, i \in N_m,$$

что влечет $x^* \in \pi(x,C+C')$. Поэтому $x \in \overline{P}^m(C+C')$. Итак, доказано, что $\forall C' \in \Omega(\varphi) \ \ \forall x \in \overline{P}^m(C) \ \ (x \in \overline{P}^m(C+C'))$, т. е. $\forall C' \in \Omega(\varphi) \ \ (P^m(C+C') \subseteq \overline{P}^m(C+C'))$ $P^{m}(C)$). Следовательно, верна оценка $\rho^{m}(C) \geq \varphi$.

Далее докажем справедливость верхней оценки. Пусть p – суть номер строки матрицы C= $[c_{ij}] \in \mathbf{R}^{m \times n}$ для которого выполнено равенство

$$||C_p||_{\infty} = \min\{||C_i||_{\infty} : i \in N_m\}.$$

Пусть $\varepsilon > ||C_p||_{\infty}$ и решение $\tilde{x} \in \overline{P}^m(C)$. Существование таких решений гарантируется нетривиальностью задачи $Z^m(C)$.

Далее построим необходимую нам возмущающую матрицу $C^* = C^*(\tilde{x})$. Пусть $C^* = C^1 + C^2$, а элементы матриц

 $C^1 = [c_{ij}^1] \in \mathbf{R}^{m \times n}, \quad C^2 = [c_{ij}^2] \in \mathbf{R}^{m \times n}$

заданы по правилам

$$c_{ij}^1 = \begin{cases} -c_{ij}, \text{ если } i=p, \ j \in N_n, \\ 0 \qquad \text{в остальных случаях}, \end{cases}$$

$$c_{ij}^2 = \tilde{x}_j \gamma_{ij}, \ (i,j) \in N_m \times N_n,$$

где

$$\gamma_{ij} = \begin{cases} (1 - \frac{1}{n})\delta, \text{ если } i = p, \ j = n, \\ -\frac{\delta}{n}, & \text{ если } i = p, \ j \neq n, \\ 0 & \text{в остальных случаях,} \end{cases}$$

$$0 < \delta < \varepsilon - ||C_p||_{\infty}.$$

Ясно, что $C^* \in \Omega(\varepsilon)$. После несложных вычислений получаем

$$(C_p + C_p^*)\tilde{x} = 0,$$

$$0 < |(C_p + C_p^*)x| \le \delta, \ x \in Q^n \setminus \{\tilde{x}, -\tilde{x}\}.$$

Откуда находим

$$|(C_p + C_p^*)\tilde{x}| < |(C_p + C_p^*)x|, \ x \in Q^n \setminus \{\tilde{x}, -\tilde{x}\}.$$

Резюмируя все выше сказанное, заключаем, что справедлива формула

$$\forall \varepsilon > ||C_p||_{\infty} \quad \forall \ \tilde{x} \in \overline{P}^m(C) \quad \exists \ C^* \in \Omega(\varepsilon) \quad \forall \ x \in Q^n \setminus \{\tilde{x}, -\tilde{x}\}$$
$$|(C_p + C_p^*)\tilde{x}| < |(C_p + C_p^*)x|.$$

Поэтому, ввиду очевидного равенства

$$|(C_p + C_p^*)\tilde{x}| = |(C_p + C_p^*)(-\tilde{x})|,$$

верно включение $\tilde{x} \in P^m(C+C^*)$, откуда и убеждаемся в справедливости верхней оценки

$$\rho^m(C) \le ||C_p||_{\infty}.$$

Наконец, докажем, что равенство $\rho^m(C)=\varphi^m(C)$ имеет место при $P^m(C)=\{x^0,-x^0\}$. В этом случае величина $\varphi^m(C)$, задаваемая формулой (2), принимает вид

$$\varphi^{m}(C) = \min_{x \in Q^{n} \setminus \{x^{0}, -x^{0}\}} \min_{i \in N_{m}} \beta_{i}(x, x^{0}), \tag{4}$$

где

$$\beta_i(x, x^0) = \min\{\alpha_i(x, x^0, q) : q \in Q\}.$$

Поэтому с учетом ранее доказанного неравенства $\rho^m(C) \geq \varphi$ для завершения доказательства теоремы осталось показать, что $\rho^m(C) \leq \xi$, где ξ – правая часть равенства (4). Для этого достаточно доказать, что для любого числа $\varepsilon > \xi$ верна формула

$$\exists C' \in \Omega(\varepsilon) \quad \exists \ \tilde{x} \in Q^n \setminus \{x^0, -x^0\} \quad (\tilde{x} \in P^m(C + C')).$$

Согласно определению числа $\xi \geq 0$ найдутся такие $x^* \in Q^n \setminus \{x^0, -x^0\}$ и $k \in N_m$, что

$$\beta_k(x^*, x^0) = \xi. \tag{5}$$

Далее будем использовать обозначения

$$N(x^{0}, x^{*}) = |\{j \in N_{n} : x_{j}^{0} = 1 \& x_{j}^{*} = -1\}|,$$

$$M(x^{0}, x^{*}) = |\{j \in N_{n} : x_{j}^{0} = x_{j}^{*}\}|,$$

$$\sigma^{*} = \operatorname{sg} C_{k} x^{*}.$$

Легко понять, что

$$2(N(x^{0}, x^{*}) + N(x^{*}, x^{0})) = ||x^{*} - x^{0}||_{1},$$
(6)

$$2M(x^0, x^*) = ||x^* + x^0||_1. (7)$$

Для построения необходимой матрицы $C'\in\Omega(\varepsilon),\ \varepsilon>\xi,$ рассмотрим три возможных случая. Случай 1: $\alpha_k(x^*,x^0,-1)<\alpha_k(x^*,x^0,1).$ Тогда с учетом (5) справедливы неравенства

$$|C_k(x^* + x^0)| > 0,$$

$$\alpha_k(x^*, x^0, -1) = \xi < \varepsilon.$$

Поэтому, задавая элементы матрицы $C' = [c'_{ij}]_{m \times n}$ по правилу

$$c'_{ij} = \begin{cases} \sigma^*\delta, & \text{если } i = k, \ x_j^0 = 1, \ x_j^* = -1, \\ -\sigma^*\delta, & \text{если } i = k, \ x_j^0 = -1, \ x_j^* = 1, \\ 0 & \text{в остальных случаях,} \end{cases}$$

где

$$\xi < \delta < \varepsilon$$
,

имеем $||C'||_{\infty} = \delta$ и, учитывая (6), выводим

$$\sigma^*(C_k + C_k')x^0 - \sigma^*(C_k + C_k')x^* = -\sigma^*C_k(x^* - x^0) + 2\delta(N(x^*, x^0) + N(x^0, x^*)) \ge$$

$$\ge -|C_k(x^* - x^0)| + \delta||x^* - x^0||_1 > -|C_k(x^* - x^0)| + \alpha_k(x^*, x^0, -1)||x^* - x^0||_1 = 0,$$

$$\sigma^*(C_k + C_k')x^0 + \sigma^*(C_k + C_k')x^* = \sigma^*C_k(x^* + x^0) = |C_k(x^* + x^0)| > 0.$$

Это означает, что для любого $q \in Q$ справедливо неравенство

$$\sigma^*(C_k + C_k')x^0 > \sigma^*q(C_k + C_k')x^*.$$

Отсюда в силу (1) будем иметь

$$|(C_k + C_k')x^0| > |(C_k + C_k')x^*|. (8)$$

Случай 2: $\alpha_k(x^*, x^0, -1) > \alpha_k(x^*, x^0, 1)$. Тогда согласно (5) находим

$$|C_k(x^*-x^0)|>0$$
,

$$\alpha_k(x^*, x^0, 1) = \xi < \varepsilon.$$

Поэтому, конструируя матрицу $C' = [c'_{ij}]_{m \times n}$ по правилу

$$c'_{ij} = \begin{cases} -\sigma^* \delta, \text{ если } i = k, \ x_j^0 = x_j^* = 1, \\ \sigma^* \delta, \quad \text{если } i = k, \ x_j^0 = x_j^* = -1, \\ 0 \quad \text{в остальных случаях,} \end{cases}$$

где $\xi < \delta < \varepsilon$, получаем $||C'||_{\infty} = \delta$ и, воспользовавшись (7), выводим

$$-\sigma^*(C_k + C_k')x^0 - \sigma^*(C_k + C_k')x^* = -\sigma^*C_k(x^* + x^0) + 2\delta M(x^0, x^*) >$$

$$> -|C_k(x^* + x^0)| + \alpha_k(x^*, x^0, 1)||x^* + x^0||_1 = 0,$$

$$-\sigma^*(C_k + C_k')x^0 + \sigma^*(C_k + C_k')x^* = \sigma^*C_k(x^* - x^0) = |C_k(x^* - x^0)| > 0$$

Итак, верны неравенства

$$-\sigma^*(C_k + C_k')x^0 > \sigma^*q(C_k + C_k')x^*, \ q \in Q.$$

которые благодаря (1) приводят к (8).

Случай 3: $\alpha_k := \alpha_k(x^*, x^0, -1) = \alpha_k(x^*, x^0, 1) = \beta_k(x^*, x^0)$. Рассмотрим два варианта. Пусть сначала $\alpha_k = 0$. Тогда

$$C_k x^0 = C_k x^* = 0. (9)$$

Легко видеть (ввиду $x^* \neq \pm x^0$), что можно выбрать два индекса $s, p \in N_n$ с условием

$$x_s^* = x_s^0, \quad x_p^* \neq x_p^0.$$

Поэтому, задавая элементы матрицы $C' = [c'_{ij}]_{m \times n}$ по правилу

$$c'_{ij} = \begin{cases} x_s^0 \delta, \text{ если } (i,j) = (k,s), \\ x_p^0 \delta, \text{ если } (i,j) = (k,p), \\ 0 \quad \text{в остальных случаях,} \end{cases}$$

где $0 = \xi < \delta < \varepsilon$, убеждаемся, что $||C'||_{\infty} = \delta$ и, принимая во внимание (9), вновь получаем неравенство (8).

Пусть теперь $\alpha_k > 0$. Тогда, повторяя все рассуждения случая 1, получаем (8).

Итак, в каждом из трех рассмотренных случаев построенна такая возмущающая матрица $C' \in \Omega(\varepsilon)$, что справедливо неравенство (8), которое свидетельствует, что $x^0 \notin \pi(x^*, C + C')$. Если $\pi(x^*, C + C') = \emptyset$, то $x^* \in P^m(C + C')$. Если $\pi(x^*, C + C') \neq \emptyset$, то благодаря внешней устойчивости множества $P^m(C + C')$ (см., например, [25]) найдется такое решение $\tilde{x} \in \pi(x^*, C + C')$, что $\tilde{x} \in P^m(C + C')$.

Таким образом, в случае, когда $P^m(C) = \{x^0, -x^0\}$, для любого числа $\varepsilon > \xi$ может быть построена такая матрица $C' \in \Omega(\varepsilon)$, что найдется решение $\hat{x} \neq \pm x^0$ с условием $\hat{x} \in P^m(C + C')$, т. е. $P^m(C + C') \not\subseteq P^m(C)$. Следовательно, $\rho^m(C) < \varepsilon$ для любого $\varepsilon > \xi$. Это и означает справедливость оценки $\rho^m(C) \le \xi$.

Теорема доказана.

4. Следствия

Следствие 1. Если нетривиальная задача $Z^m(C)$ не является устойчивой, то $\varphi^m(C) = 0$. Введем множество Слейтера $Sl^m(C)$, т. е. множество слабо эффективных решений задачи $Z^m(C)$, по правилу

$$x \in Sl^m(C) \Leftrightarrow \{x' \in Q^n : |C_i x| > |C_i x'|, i \in N_m\} = \emptyset.$$

Очевидно, что включение $P^m(C) \subseteq Sl^m(C)$ справедливо при любой матрице $C \in \mathbf{R}^{m \times n}$.

Следствие 2. При выполнении равенства $P^m(C) = Sl^m(C)$ задача $Z^m(C)$ устойчива.

Поскольку $P^{1}(C) = Sl^{1}(C)$, то следствие 2 влечет

Следствие 3. Скалярная задача разбиения $Z^1(C)$ устойчива при любом векторе $C \in \mathbf{R}^n$.

Список литературы

- 1. Сергиенко И. В., Козерацкая Л. Н., Лебедева Т. Т. Исследование устойчивости и параметрический анализ дискретных оптимизационных задач. Киев: Наукова думка, 1995. 170 с.
- 2. Гордеев Э. Н., Леонтьев В. К. Общий подход к исследованию устойчивости решений в задачах дискретной оптимизации // Журн. вычисл. математики и мат. физики. 1996. Т. 36. 1. С. 66 72.
- 3. Emelichev V. A., Girlich E., Nikulin Yu. V., Podkopaev D.P. Stability and regularization of vector problems of integer linear programming // Optimization. -2002.-V. 51. -N 4. -P. 645 676.
- 4. Сергиенко И. В., Шило В. П. Задачи дискретной оптимизации. Проблемы, методы решения, исследования. Киев: Наукова думка, 2003. 261 с.
- 5. Сотсков Ю. Н., Сотскова Н. Ю. Теория расписаний. Системы с неопределенными числовыми параметрами. Минск: НАН Беларуси, 2004. 290 с.
- 6. Лебедева Т. Т., Семенова Н. В., Сергиенко Т. И. Устойчивость векторных задач целочисленной оптимизации: взаимосвязь с устойчивостью множеств оптимальных и неоптимальных решений // Кибернетика и системный анализ. -2005. -4. C. 90-100.
- 7. Greenderg N. J. An annotated bibliography for post-solution analysis in mixed integer and combinatorial optimization // Advances in Computational and Stochastic Optimization. Logic Programming and Heuristic Search. Boston: Kluwer Acad. Publ. 1998. P. 97 148.
- 8. Tanino T., Sawaragi Y. Stability of nondominated solutions in multicriteria decision-making // Journal of Optimization Theory and Applications. 1980. V. 30. N 2. P. 229 253.

- 9. Белоусов Е. Г., Андронов В. Г. Разрешимость и устойчивость задач полиномиального программирования. М.: Изд-во МГУ, 1993. 172 с.
- 10. Sotskov Yu. N., Leontev V. K., Gordeev E. N. Some concepts of stabilty analysis in combinatorial optimization // Discrete Appl. Math. 1995. V. 58. N 2. P. 169 190.
- 11. Chakravarti N., Wagelmans A. Calculation of stability radius for combinatorial optimization problems // Operations Research Letters. 1998. V. 23. N. 1. P. 1 7.
- 12. Libura M., van der Poort E. S., Sierksma G., van der Veen J. A. A. Stability aspects of the traveling salesman problem based on k-best solutions // Discrete Appl. Math. 1998. V. 87. P. 159 185.
- 13. Hoesel S., Wagelmans A. On the complexity of postoptimality analysis of 0-1 programs // Discrete Appl. Math. 1999. V. 91. P. 251 263.
- 14. Гордеев Э. Н. Исследование устойчивости в оптимизационнных задачах на матроидах в метрике l_1 // Кибернетика и системный анализ. 2001. 2. С. 132 144.
- 15. Емеличев В. А., Кузьмин К. Г. О радиусе устойчивости эффективного решения одной векторной задачи булева программирования в метрике l_1 // Доклады РАН. 2005. Т. 401. 6. С. 733 735.
- 16. Емеличев В. А., Бердышева Р. А. О радиусе устойчивости, квазиустойчивости и стабильности векторной траекторной задачи лексикографической оптимизации // Дискретная математика. 1998. Т. 10, вып. 1. С. 20-27.
- 17. Бухтояров С. Е., Емеличев В. А. Параметризация принципа оптимальности ("от Парето до Слейтера") и устойчивость многокритериальных траекторных задач // Дискрет. анализ и исслед. операций. Сер. 2. -2003. Т. 10. 2. С. 3 18.
- 18. Бухтояров С. Е., Емеличев В. А., Степанишина Ю. В. Вопросы устойчивости векторных дискретных задач с параметрическим принципом оптимальности // Кибернетика и системный анализ. $-2003.-4.-\mathrm{C}.~155-166.$
- 19. Емеличев В. А., Кузьмин К. Г., Леонович А. М. Устойчивость в векторных комбинаторных задачах оптимизации // Автоматика и телемеханика. -2004. -2. -C. 79 -92.
- 20. Емеличев В. А., Кричко В. Н. Формула радиуса устойчивости векторной l_{∞} -экстремальной траекторной задачи // Дискретная математика. 2004. Т. 16, вып. 1. С. 14 20.
- 21. Емеличев В. А., Кузьмин К. Г. Конечные коалиционные игры с параметрической концепцией равновесия в условиях неопределенности // Известия РАН. Теория и системы управления. 2006. 2. С. 96 101.
- 22. Бухтояров С. Е., Емеличев В. А. Мера устойчивости конечной коалиционной игры с параметрическим ("от Парето до Нэша") принципом оптимальности // Журнал вычислительной математики и математической физики. 2006. Т. 46. 7, с. 1274 1280.
- 23. Танаев В. С., Гордон В. С., Шафранский Я. М. Теория расписаний. Одностадийные системы. М.: Наука, 1984.-384 с.
- 24. Lawler E. L., Lenstra J. K., Rinnoy Kan A. H. G., Shmoys D. B. Sequencing and scheduling: Algorithms and complexity // Handbook of Operations Research. Amsterdam. 1993. V. 4. P. 445 452.
- 25. Подиновский В. В, Ногин В. Д. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982. 256 с.

О верхней оценке R-различающих и разделяющих последовательностей для наблюдаемых автоматов

Евтушенко Н. В.,

Зав. Кафедрой Томского государственного университета г.Томск, 634050, пр. Ленина, 36 e-mail: yevtushenko@elefot.tsu.ru

Спицына Н. В.,

Доцент Томского государственного университета г.Томск, 634050, пр. Ленина, 36 e-mail: snv@kitidis.tsu.ru

Введение

При тестировании дискретных систем возникает задача построения тестов для недетерминированных автоматов относительно различных отношений конформности, или соответствия, между эталонным и проверяемым автоматами. При удаленном тестировании, когда для управления и наблюдения доступны каналы только одной из взаимодействующих компонент, единственным отношением, проверку которого можно гарантировать, является отношение неразделимости между недетерминированными автоматами. В этом случае, реакции двух автоматов на любую входную последовательность пересекаются [3]. Как было показано в [4], тесты, полные относительно других отношений конформности, а именно отношений эквивалентности и редукции, не являются таковыми относительно неразделимости. Кроме того, в работах [1, 4] показано, что длина разделяющей последовательности в худшем случае экспоненциально зависит от произведения числа состояний разделяемых автоматов. В данной работе мы продолжаем исследование отношения неразделимости и г-различимости между недетерминированными автоматами и показываем, что несмотря на то, что длина каждой последовательности в r-различающем множестве не превышает произведения пт числа состояний, количество последовательностей экспоненциально зависит от числа пт, причем экспоненциальная оценка является достижимой. Кроме того, мы показываем, что достижима верхняя оценка длины кратчайшей разделяющей последовательности 2^{nm-1}. Иными словами, мы показываем, что для любых целых чисел *n* и *m*, *n*,*m*≥1, существуют два наблюдаемых автомата с n и m состояниями, для которых мощность r-различающего множества равна 2^{nm-2} , и существуют два наблюдаемых автомата с n и m состояниями, для которых кратчайшая разделяющая последовательность имеет длину 2^{nm-1} .

Основные понятия и определения

Недетерминированным (ид-) автоматом A далее называется пятерка (S, I, O, h, s_1) , где S – множество состояний с выделенным начальным состоянием s_1 , I и O – соответственно входной и выходной алфавиты, $h \subseteq S \times I \times S \times O$ – отношение переходов-выходов. Элементами множества h являются четверки вида (s, i, s', o), называемые nepexodamu; при этом говорят, что автомат может перейти из состояния $s \in S$ под действием входного символа $i \in I$ в состояние $s' \in S$ с выдачей выходного символа $o \in S$, если четверка $oldsymbol{(s, i, s', o)}$ содержится в $oldsymbol{h}$.

Если для любых $(s,i,o)\in S\times I\times O$ в нд-автомате существует не более одного перехода из состояния s под действием входного символа i с выходным символом o, то говорят, что нд-автомат является naблюдаемым. Если для каждой пары $(s,i)\in S\times I$ существует хотя бы одна пара $(s',o)\in S\times O$, такая что $(s,i,s',o)\in h$, то нд-автомат называется nonhocmho onpedenehhhm.

Обычным образом отношение переходов-выходов h нд-автомата A распространяется на множество I^* входных и O^* выходных последовательностей. Пусть $s, s' \in S, \quad \alpha = i_1 i_2 ... i_k \in I^*, \beta = o_1 o_2 ... o_k \in O^*$. Тогда $(s, \alpha, s', \beta) \in h$, если существует последовательность состояний $s_1 = s, s_2, ..., s_k, s_{k+1} = s'$ таких, что $(s_i, i_i, s_{i+1}, o_i) \in h, i = 1, ..., k$. Обозначим $out(s, \alpha) = \{\beta: \exists s' \in S \ [(s, \alpha, s', \beta) \in h]\}$, т. е. $out(s, \alpha)$ есть множество выходных реакций автомата в состоянии s на входную последовательность α .

Пусть $A=(S,\,I,\,O,\,h,\,s_1),\,B=(T,\,I,\,O,\,g,\,t_1),\,s\in S,\,t\in T,\,$ - полностью определенные автоматы. Автомат B называется *подавтоматом* автомата A, если $T\subseteq S,\,t_1=s_1$ и $g\subseteq h$. *Пересечением* автоматов $A=(S,\,I,\,O,\,h,\,s_1)$ и $B=(T,\,I,\,O,\,g,\,t_1)$ (обозначение $A\cap B$), назовем максимальный связный подавтомат инициального автомата $(S\times T,\,I,\,O,\,H,\,s_1t_1),\,$ в котором функция поведения $(st,\,i,\,s't',\,o)\in H\Leftrightarrow (s,\,i,\,s',\,o)\in h$ & $(t,\,i,\,t',\,o)\in g$. Пересечение автоматов описывает общую часть поведения автоматов A и B и используется для построения входных последовательностей, различающих эти автоматы.

Состояние t автомата B называется $pe \partial y \kappa u u e u$ состояния s автомата A, если $\forall \alpha \in I^*$ [$out(t, \alpha) \subseteq out(s, \alpha)$], т.е. если для любой входной последовательности множество выходных последовательностей автомата B содержится во множестве выходных последовательностей автомата A [2]. Если состояние t_1 является редукцией состояния s_1 , то автомат B называется $pe \partial y \kappa u u e u$ автомата A.

Состояние s автомата A и состояние t автомата B неразделимы, если $\forall \alpha \in I^*$ [$out(s, \alpha) \cap out(t, \alpha) \neq \emptyset$]. Если $\exists \alpha \in I^*$ [$out(s, \alpha) \cap out(t, \alpha) = \emptyset$], то состояния s и t разделимы no α , или просто разделимы [2]. Автоматы A и B неразделимы, если $s_1 \sim t_1$. Если $s_1 \nsim_{\alpha} t_1$, то автоматы A и B разделимы no a, или просто разделимы; последовательность a называется разделяющей последовательностью автоматов a и a. Таким образом, автоматы разделимы, если существует входная последовательность, для которой множества выходных последовательностей автоматов не пересекаются. Разделяющая последовательность a a0 a1 называется a1 называется a2 a3 не короче a4. Если автоматы неразделимы, то для любой входной последовательности множества выходных последовательностей автоматов пересекаются.

Построение г-различающего множества

Множество W(st) входных последовательностей, называемое r-различающим множеством состояний s и t, строится итеративно следующим образом. Сначала находятся все пары s', t' 1-r-различимых состояний, и соответствующие символы $i \in I$ [$out(s',i) \cap out(t',i) = \varnothing$] заносятся в множество W_1 , т.е. $W_1 = \bigcup W_1(s't')$. Пусть уже определено множество W_{k-1} для некоторого k > 1. Тогда $W_k = \bigcup W_k(s''t'')$, где s''t'' - пары всех k-r-различимых состояний, $W_k(s''t'') = \{iW_j(s't'): \exists o \in O(s''t'', i, s't', o) \in H, j \ge 1\}$.

Очевидно, что всякая r—различающая последовательность имеет длину не более nm, где n и m — число состояний автоматов A и B, соответственно. Последовательность такой длины возникает в случае, когда для любого k существует единственная пара k–r-различимых состояний. Особый интерес представляет верхняя оценка суммы длин последовательностей r-различающего множества.

Верхняя оценка суммы длин последовательностей г-различающего множества

Теорема 1. Для заданных целых чисел n u m, $n \ge 1$, $m \ge 1$, всегда найдутся r-различимые автоматы A u B c числом состояний n u m, соответственно, такие что для них сумма длин последовательностей

$$r$$
-различающего множества $l_{nm} = 2^{nm-2} + \sum_{i=1}^{nm-1} l_i$.

Доказательство. Приведем конструктивное доказательство теоремы, т.е. покажем, каким образом построить такие автоматы $A = (S, I, O, h, s_1)$ и $B = (T, I, O, g, t_1)$ с числом состояний n и m, соответственно,

для которых сумма длин последовательностей r-различающего множества $l_{nm} = 2^{nm-2} + \sum_{i=1}^{nm-1} l_i$. Входной алфавит автоматов $I = \{i_1, ..., i_{nm}\}$, выходной алфавит автоматов $O = \{0, 1\} \times S \times T$.

Будем перебирать определенным образом подмножества множества $S \times T \setminus s_1 t_1 = \{s_1 t_2, ..., s_n t_m\}$. Подмножество $M_1 = (S \times T) \setminus s_1 t_1$, подмножество M_k получается из подмножества M_{k+1} путем удаления произвольного состояния (кроме $s_n t_m$), т.е. последнее подмножество $M_{nm-1} = \{s_n t_m\}$. Для каждого входного символа отношения h и g содержат переходы, описанные ниже. Множества переходов по входным символам i_1 и i_{nm} отличаются от переходов по остальным входным символам.

Переходы по входному символу i_1 : Для входного символа i_1 отношение h содержит все переходы из множества $\{(s_1, i_1, (1, s, t), s): s \in S, t \in T, st \neq s_1t_1\}$, и отношение g содержит все переходы из множества $\{(t_1, i_1, (1, s, t), t): s \in S, t \in T, st \neq s_1t_1\}$. Кроме того, для каждого состояния $s \in S \setminus s_1$ и $t \in T \setminus t_1$, отношение h содержит переходы $(s, i_1, (0, s_1, t_1), s_1)$ и $(s, i_1, (1, s, t_1), s_1)$ и отношение g содержит переходы $(t, i_1, (0, s_1, t_1), t_1)$ и $(t, i_1, (1, s_1, t), t_1)$.

Переходы по входному символу i_{nm} : Входной символ i_{nm} должен 1-r-различать состояния s_n и t_m ; поэтому, отношение h содержит переход $(s_n, i_{nm}, (1, s_1, t_1), s_1)$ и отношение g содержит переход (t_m, i_{nm}, t_1, t_2)

 $(0,s_1,t_1), t_1)$. Кроме того, для каждого состояния $s \in S \setminus s_n$ и $t \in T \setminus t_m$, отношение h содержит переходы $(s, i_{nm}, (0,s_1,t_1)), s_1)$ и $(s, i_{nm}, (1,s_1,t_1)), s_1)$ и отношение g содержит переходы $(t, i_{nm}, (0,s_1,t_1)), t_1)$ и $(t, i_{nm}, (1,s_1,t_1)), t_1)$.

Переходы по входному символу i_{k+1} , $1 \le k \le nm-2$: Для входного символа i_{k+1} , $1 \le k \le nm-2$, отношения переходов-выходов h и g образованы на основе подмножеств M_k и M_{k+1} , построенных описанным выше способом. Для состояний $s \in S$ и $t \in T$ таких, что пара $st \in M_k$ и $st \notin M_{k+1}$ отношение h содержит все переходы из множества $\{(s, i_{k+1}, (1, s', t'), s'): s't' \in M_{k+1}\}$ и $\{(s'', i_{k+1}, (0, s'', t''), s_1): s'' \neq s \lor t'' \neq t\}$, отношение g содержит все переходы множества $\{(t, i_{k+1}, (1, s', t'), t'): s't' \in M_{k+1}\}$ и $\{(t'', i_{k+1}, (0, s'', t''), t_1): s'' \neq s \lor t'' \neq t\}$.

По построению, для любого состояния s автомата A (состояния t автомата B) и любого входного символа i не существует переходов из состояния s (t) в различные состояния с одинаковым выходным символом, поэтому автоматы A и B наблюдаемые.

Для автоматов A и B, построенных описанным выше способом, единственная пара (1-r)-различимых состояний — это состояния s_n и t_m ($M_{nm-1} = \{s_n t_m\}$), $W_1 = \{i_{nm}\}$. По построению, множество $M_{nm-2} = M_{nm-1} \cup \{st\} = \{s_n t_m, st\}$, где $st \in (S \times T) \setminus \{s_1 t_1, s_n t_m\}$. Поэтому единственная пара 2-r-различимых состояний — это состояния s и t, $W_2 = \{i_{nm-1} i_{nm}\}$. Множество $M_{nm-3} = M_{nm-2} \cup \{st\}$, где $st \in (S \times T) \setminus (\{s_1 t_1\} \cup M_{nm-2})$. Поэтому единственная пара 3-r-различимых состояний — это состояния s и t, $W_3 = \{i_{nm-2} i_{nm}, i_{nm-2} i_{nm-1} i_{nm}\}$. Таким образом, структура построенных описанным выше способом наблюдаемых автоматов A и B такова, что из k-r-различимых состояний, которые не являются (k-1)-r-различимыми, по входному символу i_{nm-k+1} в автомате $A \cap B$ есть переходы во все j-r-различимые состояния, где j < k. Поэтому, для автоматов A и $B \mid W_1 \mid = 1, \mid W_2 \mid = 1, \mid W_k \mid = 2 \cdot \mid W_{k-1} \mid$ для k > 2, и, следовательно, $\mid W_{nm} \mid = 2^{nm-2}$. При построении множества W_k по множествам W_1 , ..., W_{k-1} ко всем последовательностям этих множеств будет добавлено по одному символу, т.е. сумма длин

последовательностей множества
$$W_k$$
 составит $l_k = \sum_{i=1}^{k-1} |W_i| + \sum_{i=1}^{k-1} l_i = 2^{k-2} + \sum_{i=1}^{k-1} l_i$.

Верхняя оценка длины кратчайшей разделяющей последовательности

В работах [1, 4] показано, что длина кратчайшей разделяющей последовательности в худшем случае экспоненциально зависит от произведения числа состояний разделяемых автоматов. Получено конструктивное доказательство достижимости верней оценки, т.е. показано, какую структуру имеют наблюдаемые автоматы с n и m состояниями, для которых кратчайшая разделяющая последовательность имеет длину 2^{nm-1} .

Теорема 2. Для заданных целых чисел n u m, $n \ge 1$, $m \ge 1$, всегда найдутся разделимые автоматы A u B c числом состояний n u m, соответственно, такие что для них кратчайшая разделяющая последовательность имеет длину 2^{nm-1} .

Доказательство теоремы 2 можно найти в работе [5].

Заключение

В данной работе исследуются отношения неразделимости и r-различимости между недетерминированными автоматами. Показывается, что верхняя оценка длины кратчайшей разделяющей последовательности 2^{nm-1} является достижимой. Кроме того, показывается, что несмотря на то, что длина каждой последовательности в r-различающем множестве не превышает произведения nm числа состояний, количество последовательностей экспоненциально зависит от числа nm, причем экспоненциальная оценка является достижимой.

Список литературы.

- 1. R.Alur, C.Courcoubetis, M.Yannakakis. Distinguishing tests for nondeterministic and probabilistic machines // Proceedings of the 27th ACM Symposium on Theory of Computing, 1995, P. 363-372.
- 2. Евтушенко Н. В., Куфарева И. Б. Отношения между недетерминированными автоматами. Томск: Изд-во Института оптики атмосферы СО РАН, 2001 26 с.
- 3. Спицына Н. В. Синтез тестов для проверки взаимодействия дискретных управляющих систем методами теории автоматов: Диссертация на соискание ученой степени канд. технических наук. Томск, 2005. 158 с.
- 4. Евтушенко Н. В., Спицына Н. В. Исследование отношения неразделимости для недетерминированных автоматов // Вестник ТГУ. Приложение. Серия «Математика. Кибернетика. Информатика». 2005. № 14. С. 218–222.
- 5. Евтушенко Н. В., Спицына Н. В. О верхней оценке длины разделяющей последовательности // Вестник ТГУ. Приложение. Серия «Математика. Кибернетика. Информатика». 2006 (в печати).

К вопросу об обфускации конечных автоматов

Захаров В. А., E-mail: zakh@cs.msu.su ф-т ВМиК МГУ, доцент

Обфускация программы (от лат. «obfuscatio» — затемнение) — это метод информационной защиты программы путем преобразование ее к такому виду, при котором исходная функциональность программы полностью сохраняется, но извлечение из кода обфускированной программы информации об используемом алгоритме и структурах данных, а также внесение в программу целенаправленных изменений становится вычислительно сложной задачей. Тем самым информационная защита программы обеспечивается специальной формой представления программы, благодаря которой программный код становится очень трудным для изучения и модификации.

Идея обфускации программ и ее приложения в криптографии были предложены в работе Диффи и Хеллмана [4] в 1976 г., заложившей математические основы современной криптографии. Двадцать лет спустя проблема обфускации программ была поставлена в работе [3] в связи с разработкой новых методов информационной защиты программного обеспечения.

Интерес к задаче построения стойких обфускирующих преобразований программ обусловлен многими факторами. Стойкая обфускация программ позволяет преобразовывать криптосистемы с секретным ключом в криптосистемы с открытым ключом, защищать алгоритмы как объекты интеллектуальной собственности, обеспечивать информационную безопасность мобильных агентов. Это лишь краткий перечень полезных приложений обфускации. Применение обфускации может привести также и к негативным последствиям: с ее помощью открывается возможность создания трудно обнаруживаемых компьютерных вирусов, программных «закладок» и пр. Более подробно о математических и программистских аспектах обфускации рассказывается в работах [1,3]) Следует отметить также, что проблема обфускации затрагивает целый ряд вопросов, связанных с моделированием интеллектуальной деятельности. Стойкость обфускации подразумевает, что по описанию (коду) программы очень трудно догадаться о свойствах алгоритма, реализованного этой программой. Это вынуждает искать ответы на такие фундаментальные вопросы, как «В чем проявляется познаваемость программы?», «Как сравнивать по уровню познаваемости две программы?», «Существуют ли непознаваемые программы?».

В настоящее время имеется совсем немного значимых результатов исследования проблемы обфускации. В работе [2] было приведено первое формальное определение стойкости обфускации и доказана теорема о невозможности построения стойких обфускаторов в некоторых классах программ. Эти результаты были усилены в работах [5,8]. Вместе с тем в работах [6,7,9] было показано, что стойкая (при стандартных криптографических предположениях) обфускация возможна для некоторых программ специального вида. Разрыв между положительными и отрицательными результатами велик: для подавляющего большинства практически используемых программ вопрос об их стойкой обфускируемости остается открытым. Для многих моделей вычисления задача обфускации вообще не рассматривалась. В частности, неизученным остается вопрос об обфускации конечных автоматов.

В настоящей заметке приводится одно из возможных формальных определений обфускации конечных автоматов, а также описание одного метода обфускации.

Мы будем рассматривать неинициальные детерминированные конечные автоматы Мили, которые задаются пятерками вида $\mathcal{A}=\langle \Sigma,\Delta,Q,\varphi,\psi\rangle$, где Σ,Δ — входной и выходной алфавиты; Q — множество состояний, $\varphi:Q\times\Sigma\to Q$ — функция перехода, а $\psi:Q\times\Sigma\to\Delta$ — функция выхода. Поскольку обфускации подвергается программа (схема), являющаяся реализацией автомата, мы будем считать, что

- элементы алфавитов Σ и Δ , а также состояния множества Q представлены двоичными наборами длины $k = |\log \Sigma|$ [, $m = |\log \Delta|$ и $n = |\log Q|$ соответственно;
- функция перехода φ представлена набором, состоящим из n булевых формул $\varphi^i(x_1,\ldots,x_n,y_1,\ldots,y_k),\ 1\leq i\leq n,$ вычисляющих для каждого состояния $q\in Q$ и входной буквы $a\in A$ биты состояния $\varphi(q,a)$;
- функция выхода ψ представлена набором, состоящим из m булевых формул $\psi^j(x_1,\ldots,x_n,y_1,\ldots,y_k),\ 1\leq j\leq m,$ вычисляющих для каждого состояния $q\in Q$ и входной буквы $a\in A$ биты выходной буквы $\psi(q,a)$.

Таким образом, автомат \mathcal{A} реализуется *программой* $S_{\mathcal{A}}$, состоящей из наборов переменных $\mathbf{x} = \langle x_1, \dots, x_n \rangle$, $\mathbf{y} = \langle y_1, \dots, y_k \rangle$ и двух наборов булевых формул $\Phi = \langle \varphi^1, \dots, \varphi^n \rangle$ и $\Psi = \langle \psi^1, \dots, \psi^m \rangle$. В

дальнейшем мы будем предполагать, что параметры k и m одинаковы для всех рассматриваемых автоматов, а размер формул в программе каждого автомата \mathcal{A} ограничен величиной $O(|Q|^2)$. Размером $|S_{\mathcal{A}}|$ программы автомата считается суммарный размер всех формул, входящих в ее состав.

Инициальный автомат — это конечный автомат \mathcal{A}_q , в котором выделено начальное состояние $q \in Q$. Его программой является пара $(S_{\mathcal{A}},q)$. Такое разделение автомата и его начального состояния объясняется тем, что в языках Verilog и VHDL описание автомата состоит из двух блоков: блока инициализации (оператор initial), который устанавливает начальное состояние автомата, и блока вычисления (оператор always), в котором описана программа автомата. Две программы $(S_{\mathcal{A}},q)$ и $(S_{\mathcal{B}},p)$ считаются эквивалентными, если соответствующие этим программам конечные инициальные автоматы \mathcal{A}_q и \mathcal{B}_p вычисляют одну и ту же ограниченно-детерминированную функцию.

Обфускатором конечных автоматов будет называться такой (вероятностный) алгоритм \mathcal{O} , который преобразует всякую программу $(S_{\mathcal{A}},q)$ инициального автомата \mathcal{A}_q в программу $(\mathcal{O}(S_{\mathcal{A}}),\mathcal{O}(q))=(S_{\mathcal{D}},p)$ другого инициального автомата \mathcal{D}_p и удовлетворяет при этом следующим требованиям:

- 1. ϕ ункциональность: программы $(S_{\mathcal{A}},q)$ и $(\mathcal{O}(S_{\mathcal{A}}),\mathcal{O}(q))$ эквивалентны;
- 2. эффективность: существует такой полином poly, что для почти всех программ автоматов $S_{\mathcal{A}}$ выполняется неравенство $|\mathcal{O}(S_{\mathcal{A}})| \leq poly(|S_{\mathcal{A}}|)$;
- 3. cmoйкость: для любой полиномиальной вероятностной машины Тьюринга M существует такая пренебрежимо малая функция $\mu(\cdot)$, что для программ любых двух автоматов $\mathcal A$ и $\mathcal B$, имеющих N состояний каждый, выполняется неравенство

$$|Prob[M(\mathcal{O}(S_{\mathcal{A}}), \mathcal{A}, \mathcal{B}) = 1] - Prob[M(\mathcal{O}(S_{\mathcal{B}}), \mathcal{A}, \mathcal{B}) = 1]| \le \mu(N),$$

где вероятность определяется случайными величинами, используемыми вероятностными алгоритмами \mathcal{O} и M.

Стойкость обфускации означает, что без знания начального состояния $\mathcal{O}(q)$ (которое играет здесь роль секретного ключа) практически невозможно понять, какой из двух инициальных конечных автоматов подвергся обфускации. В качестве примера рассмотрим один метод обфускации конечных автоматов и оценим, в какой мере он удовлетворяет указанному определению.

Пусть задана программа конечного автомата $S_{\mathcal{A}} = (\mathbf{x}, \mathbf{y}, \Phi, \Psi)$, имеющего множество состояний Q, и начальное состояние q_0 , и пусть выбрано произвольное натуральное число r. Предлагаемое нами преобразование программ конечных автоматов \mathcal{O}_r работает следующим образом:

- 1) Выбираются случайным образом r+1 перестановок ξ_i , $i=\overline{0,r}$, множества состояний Q, а также r вектор-функций Ψ_i , $i=\overline{1,r}$, размерности $m=]\log |\Delta|[$ и r(r+1) вектор-функций Φ_{ij} , $i=\overline{1,r}$, $j=\overline{0,r}$ размерности $n=]\log |Q|[$.
- 2) Формируется набор из (r+1)n переменных $\mathbf{z} = \langle \mathbf{z}^0, \dots, \mathbf{z}^r \rangle$, разбитый на r+1 блоков \mathbf{z}^i по n переменных в каждом блоке.
 - 3) Вектор функции Ψ_0 и Φ_{0j} , $j=\overline{0,r}$, определяются следующим образом:

$$\Psi_0(\mathbf{z}^0, \mathbf{y}) = \Psi(\xi_0^{-1}(\mathbf{z}^0), \mathbf{y}) \oplus \bigoplus_{i=1}^r \Psi_i(\xi_i(\xi_0^{-1}(\mathbf{z}^0)), \mathbf{y}),$$

$$\Phi_{0j}(\mathbf{z}^0, \mathbf{y}) = \xi_j(\Phi(\xi_0^{-1}(\mathbf{z}^0), \mathbf{y})) \oplus \bigoplus_{i=1}^r \Phi_{ij}(\xi_i(\xi_0^{-1}(\mathbf{z}^0)), \mathbf{y}).$$

4) Формируются (асимптотически наилучшим методом) формулы для вычисления векторфункций

$$\Psi' = \bigoplus_{i=0}^r \Psi_i(\mathbf{z}^i, \mathbf{y})$$

И

$$\Phi' = \langle \bigoplus_{i=0}^r \Phi_{i0}(\mathbf{z}^i, \mathbf{y}), \dots, \bigoplus_{i=0}^r \Phi_{ir}(\mathbf{z}^i, \mathbf{y}) \rangle.$$

5) Результатом обфускации объявляется программа автомата $\mathcal{O}_r(S_{\mathcal{A}}) = (\mathbf{z}, \mathbf{y}, \Phi', \Psi')$ с начальным состоянием $\mathcal{O}_r(q_0) = \langle \xi_0(q_0), \xi_1(q_0), \dots, \xi_r(q_0) \rangle$.

Как видно из описания этого преобразования, полученная в результате обфускации программа автомата $\mathcal{O}_r(S_{\mathcal{A}})$ состоит из r+1 блоков S_0, S_1, \ldots, S_r . В каждом блоке $S_i, 0 \leq i \leq r$, вычисляются вектор-функции Ψ_i и $\Phi_{i0}, \Phi_{i1}, \ldots, \Phi_{ir}$. Функции перехода и выхода результирующего автомата представляют собой композиции функций, вычисляемых этими блоками.

Для описанного преобразования программ конечных автоматов \mathcal{O}_r справедливы следующие утверждения.

Теорема 1. Для любой программы (S_A, q) конечного инициального автомата программы (S_A, q) и $(\mathcal{O}_r(S_A), \mathcal{O}_r(q))$ эквивалентны.

Теорема 2. Существует такая константа C, что для почти всех программ автоматов S_A выполняется неравенство $|\mathcal{O}_r(S_A)| \leq C \cdot (r+1)^2 \cdot |S_A|^2$.

Вопрос о том, является ли описанное нами преобразование конечных автоматов стойкой обфускацией, остается открытым. Но можно привести доводы в пользу того, что анализ преобразованных таким образом автоматов является трудной задачей.

Условимся, что для преобразованной описанным выше способом программы конечного автомата $\mathcal{O}_r(S_{\mathcal{A}})$ запись $\mathcal{O}^t(S_{\mathcal{A}})$, где $0 \leq t \leq r$, будет обозначать набор формул, соответствующих векторфункциям $\Psi_i, \ i = \overline{1,r}, \ i \neq t, \$ и $\Phi_{ij}, \ i = \overline{0,r}, \ j = \overline{0,r}, \ i \neq t.$ Таким образом, $\mathcal{O}^t(S_{\mathcal{A}})$ — это часть программы обфускированного автомата, полученная удалением одного из составляющих ее блоков S_t .

Теорема 3. Для любой полиномиальной вероятностной машины Тьюринга M существует такая пренебрежимо малая функция $\mu(\cdot)$, что для программ любых двух автоматов \mathcal{A} и \mathcal{B} , имеющих N состояний каждый, и для любых целых $t_1, t_2, \ 0 \le t_1, t_2 \le r$, выполняется неравенство

$$|Prob[M(\mathcal{O}_r^{t_1}(S_{\mathcal{A}}), \mathcal{A}, \mathcal{B}) = 1] - Prob[M(\mathcal{O}_r^{t_2}(S_{\mathcal{B}}), \mathcal{A}, \mathcal{B}) = 1]| \le \mu(N),$$

где вероятность берется по случайным величинам, используемым вероятностными алгоритмами \mathcal{O}_r и M.

Теорема 3 означает, что в том случае, если хотя бы один из блоков обфускированной программы становится недоступным для анализа, то по оставшейся части программы практически невозможно понять, какой из двух инициальных конечных автоматов подвергся обфускации.

Список литературы

- 1. Варновский Н.П., Захаров В.А., Кузюрин Н.Н. Математические проблемы обфускации. Труды конференции "Математика и безопасность информационных технологий" (МаБИТ-04) (Москва, 28-29 октября $2004~\mathrm{r.}$), 2005.
- 2. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K. On the (Im)possibility of obfuscating programs. CRYPTO'01 Advances in Cryptology, Lecture Notes in Computer Science, v. 2139, 2001, p. 1–18.
- 3. Collberg C., Thomborson C., Low D., A Taxonomy of Obfuscating Transformations. Tech. Report, N 148, Univ. of Auckland, 1997.
- 4. Diffie W., Hellman M. New directions in cryptography. IEEE Transactions on Information Theory, IT-22(6), 1976, p. 644–654.
- 5. Goldwasser S., Kalai Y.T. On the impossibility of obfuscation with auxiliary inputs. In Proc. of the 46-th IEEE Symposium on Foundations of Computer Science (FOCS 2005).
- 6. Lynn B., Prabhakaran M., Sahai A. Positive Results and Techniques for Obfuscation. In Proceedings of Eurocrypt 2004.
- 7. Wee H. On obfuscating point functions. In Proc. of 37-th Symposium on Theory of Computing, 2005, p. 523–532.

- 8. Varnovsky N.P. A note on the concept of obfuscation. Труды ИСП РАН, 2004, т. 6, с. 137–149.
- 9. Zakharov V.A., Varnovsky N.P. On the possibility of provably secure obfuscating programs. Lecture Notes in Computer Science, v. 2890, 2003, p. 91–102.

Об универсальных однородных ловушках

Г. Килибарда

Технолого-металлургический факультет, Белградский университет, Карнегиева, 4, 11001 Белград, Сербия e-mail: gkilibar@EUnet.yu

Мозаичный n-мерный лабиринт называется однородной ловушкой для данного множества автоматов, если для любого автомата $\mathfrak A$ этого множества существует такое положительное число $r=r(\mathfrak A)$, что автомат $\mathfrak A$, помещенный в произвольную вершину этого лабиринта, не покидает в нем область, диаметр которой больше, чем r. Если множество автоматов включает все допустимые для мозаичных n-мерных лабиринтов автоматы, то эта ловушка называется универсальной однородной ловушкой. В работе исследутся вопрос о существовании как однородных ловушек для произвольного конечного множества автоматов, так и вопрос о существовании универсальных однородных ловушек.

Граф, у которого вершины являются целочисленными точками евклидового пространства \mathbb{R}^n и который допускает реализацию в \mathbb{R}^n , которая оставляет его вершины на своих местах и в которой его ребра представлены единичными отрезками, параллельными координатным осям, называется n-мерным мозаичным лабиринтом. Ясно, что если такая реализация существует, то она является единственной, и чтобы облегчить наши рассуждения мы можем считать, что имеем дело не с графом, а с его реализацией указанного типа. В таком случае, любому ребру, инцидентному данной вершине, вполне естественным способом можно сопоставить одно из 2n "координатных направлений" в \mathbb{R}^n . Если n=2, то вместо "2-мерный мозаичный лабиринт" говорим n-лоский лабиринти. Рассматриваются как конечные лабиринты, у которых конечное число вершин, так и бесконечные лабиринты, у которых бесконечное число вершин.

Если некоторая вершина лабиринта отмечена, которую в таком случае называем его *входом* или *начальной вершиной*, то лабиринт называется *инициальным*, в противном случае говорим, что он является *неинициальным*. Ниже будем всегда, если не сказано иначе, предполагать, что рассматриваемый лабиринт является инициальным.

Инициальный автомат "способный к передвижению" в *n*-мерных мозаичных лабиринтах называется *n*-мерным автоматом. Такой автомат, оказавшись в некотором *n*-мерном мозаичном лабиринте, в любой момент дискретного времени находится в некоторой его вершине — в *текущей вершине*. Его входной буквой является информация о направлениях ребер, инцидентных текущей вершине. В зависимости от его состояния и входной буквы, автомат выбирает одно из направлений, принадлежащих входной букве, и тем самым выбирает одно из ребер, инцидентных текущей вершине. В следующий момент он переходит в вершину, которая по выбранному ребру связана с текущей вершиной. В начальный момент автомат помещается во входную вершину данного лабиринта и начинает движение в нем со своего начального состояния. Автомат *обходит* данную вершину лабиринта, если существует момент дискретного времени, в котором он оказывается в ней. Если автомат обходит все вершины лабиринта, то говорим, что автомат *обходит* данный лабиринт. Если данный автомат не обходит лабиринт, то этот лабиринт является *ловушкой* для него. В дальнейшем все автоматы являются *n*-мерными.

Конечное множество автоматов называется *независимой системой автоматов*, если автоматы этого множества, будучи помещенными в начальный момент времени в произвольный лабиринт, не "коммуницируют" между собой и не "видят" друг друга. Лабиринт является ловушкой для независимой системы автоматов, если существует хотя бы одна вершина, которую не обходит ни один автомат данной системы. Лабиринт является *слабой ловушкой* для данной системы, если он является ловушкой для любого автомата по отдельности. Показано (см., например, [1]), что для любой (конечной) системы автоматов существует конечная плоская ловушка.

Если каждый из автоматов системы, помещенной в лабиринт, учитывает и состояния автоматов которые оказались в его текущей вершине, то говорим, что данная система автоматов является коллективом (n-мерных) автоматов. Аналогичным способом, как и выше, для коллективов автоматов вводится понятие обходимости и ловушки.

Введем некоторые обозначения, которыми мы будем пользоваться ниже.

Для данного конечного множества автоматов \mathcal{A} через [\mathcal{A}] обозначим множество всех инициальных автоматов, которые можно получить из автоматов данной системы варьированием их начальных состояний.

Будем считать, что \mathbf{R}^n — евклидовое n-мерное пространство, наделенное стандартной метрикой. Под n-мерным (открытым) шаром (кругом, если n=2) с центром в точке $x\in\mathbf{R}^n$ и радиусом $r\in\mathbf{R}^+$ понимается, как обычно, множество $\mathrm{Bl}_n(x,r)=\{y\in\mathbf{R}^n\,|\,d(x,y)< r\}$. Напомним, что некоторое множество $V\in\mathbf{R}^n$ считается ограниченным, если $V\subseteq\mathrm{Bl}_n(x,r)$ для некоторых $x\in\mathbf{R}^n$ и $r\in\mathbf{R}^+$, или, что то же самое, diam $V=\sup\{d(x,y)\,|\,x,y\in V\}< r'$ для некоторого $r'\in\mathbf{R}^+$.

Говорим, что данный автомат $\mathfrak A$ *не справляется* с данным бесконечным плоским мозаичным лабиринтом, если он, начиная свое движение с его входа, не покидает в нем какую-то ограниченную область. Ясно, что из-за "мозаичности" лабиринта это условие эквивалентно требованию, чтобы множество вершин, посещаемых автоматом, было конечно.

Пусть дана независимая система автоматов А. Условия, которые выше определяли два основных типа ловушек, в случае неинициального лабиринта можно сочетать еще и с требованием, чтобы в начальный момент все автоматы данной системы были помещены в одно (произвольное) поле лабиринта, или с требованием, чтобы в начальный момент все автоматы были разбросаны произвольным способом по лабиринту. При этом предполагается, что в случае неинициальной системы автоматов, когда у автоматов не заданы начальные состояния, они начинают работу со своего произвольного состояния. Если все это "пропустить" еще и через "призму несправляемости", а не невозможности обхода, то получаем целый спектр ловушек.

Интересно отметить, что, несмотря на кажущееся разнообразие, случаи всех этих ловушек легко разбираются с помощью ловушки "основного" типа. Так, например, нетрудно удостовериться, что если надлежащим способом связывать в "конгломерат" определенное количество копий "обычной" конечной ловушки для данной системы $[\mathcal{A}]$, можно получить конечную ловушку любого имеющего смысл типа для \mathcal{A} . Одна конструкция такого типа дана в [2].

Проблема построения бесконечной плоской мозаичной ловушки для всех конечных независимых систем автоматов фактически решена в [1]. Там было показано, что можно построить бесконечный плоский мозаичный лабиринт, с которым не справляется ни один конечный инициальный автомат. Нетрудно убедиться, что отсюда прямо следует достаточно общий результат, касающийся этой проблемы, а именно, что

Теорема 1. Существует неинициальный бесконечный плоский мозаичный лабиринт, такой, что для любой независимой системы автоматов и для любого начального расположения автоматов этой системы (для любого автомата выбирается своя начальная вершина) множество вершин, которые система обходит, является конечным.

Также известно, что в случае 3-мерных мозаичных лабиринтов не существует коллектив автоматов, обходящий все 3-мерные конечные мозаичные лабиринты. Доказательство этого факта дается в [3] и [5], и проводится посредством построения соответствующей ловушки для заданного коллектива автоматов. Более того, в [5] показано, что для любого конечного множества коллективов автоматов существует конечная 3-мерная мозаичная инициальная ловушка ограниченной "толщины" — все ее вершины лежат только в двух плоскостях.

Назовем неинициальный бесконечный мозаичный n-мерный лабиринт L (мозаичной n-мерной) универсальной ловушкой, если любой коллектив автоматов, стартуя из любого набора вершин лабиринта L, не обходит его. Показано ([9]), что так же, как и в случае независимых систем автоматов в плоскости, существует универсальная ловушка для любого $n \geq 3$ (ясно, что при этом достаточно было показать, что такая ловушка существует для n=3). Также известно, что такая плоская ловушка не существует (см., например, [3], [4] или [7]).

Назовем неинициальный бесконечный мозаичный n-мерный лабиринт L npocmoй (мозаичной n-мерной) универсальной ловушкой, если любой (n-мерный) автомат, стартуя из любой вершины лабиринта L, не обходит его. Из предыдущего абзаца и из Теоремы 1 ясно, что такая ловушка существует при любом $n \geq 2$. Поэтому интересно исследовать и более сильные варианты универсальных ловушек.

Назовем (мозаичную n-мерную) универсальную ловушку L=(V,E) однородной, если для любого коллектива \mathcal{A} , состоящего из m автоматов, существует положительное число $r=r(\mathcal{A})$, такое, что при любом начальном расположении $v_1,\ldots,v_m\in V$ автоматов данного коллектива (автоматы

упорядочены каким-то способом, и i-тый автомат в начальный момент помещен в вершину v_i) множество всех вершин, которые коллектив обходит, лежит в $\mathrm{Bl}_n(v, r + \mathrm{diam}\{v_1, \ldots, v_m\})$ при некотором $v \in \mathbf{R}^n$. В [9] показано, что существуют 3-мерные универсальные однородные ловушки.

Теперь аналогичным способом можно определить соответствующие однородные ловушки как для всех n-мерных автоматов, так и для некоторого множества таких автоматов.

Назовем неинициальный бесконечный мозаичный n-мерный лабиринт L=(V,E) однородной ловушкой для данного множества автоматов, если для любого автомата $\mathfrak A$ данного множества автоматов существует $r=r(\mathfrak A)$ такое, что при любом выборе начальной вершины $v\in V$ множество всех вершин, которые обходит автомат $\mathfrak A$, является подмножеством множества $\mathrm{Bl}_n(v,r)$. Если в качестве данного множества автоматов взять все n-мерные автоматы, то однородную ловушку называем npocmoй универсальной однородной (n-мерной) ловушкой.

Основные результаты данной работы могут быть сформулированы в виде двух следующих теорем.

Пользуясь результатами работы [10], где показано существование автоматов, обходящих плоские мозаичные лабиринты с ограниченными дырами, можно показать, что имеет место следующая теорема.

Теорема 2. Простые n-мерные однородные универсальные ловушки существуют только, если $n \ge 3$.

С другой стороны, эффективным построением соответствующей ловушки (возможность такого построения следует в конечном итоге из результатов и построений, приведенных в работах [6] и [8]), доказывается следующая теорема.

Теорема 3. В плоскости для любого конечного множества автоматов существует однородная ловушка.

Список литературы

- 1. Antelmann H., Budach L., Rollik H. A. On universale traps // EIK. - 1979. - Vol. 15, no. 3. - Pp. 123–131.
- 2. Килибарда Γ . Об универсальных лабиринтах-ловушках для конечных множеств автоматов // Дискретная математика. 1990. Т. 2., вып. 1. С. 72—79.
- 3. Blum M., Sakoda W. On the capability of finite automata in 2 and 3 dimensional space // Proc. 17th IEEE FOCS Conf. 1977. Pp. 147–161.
- 4. Habasinski Z., Karpinski M. A codification of Blum-Sakoda 7-pebbles algorithm // ICS PAS Reports 448. — Warszawa, 1981.
- 5. Hemmerling A. Normed two-plane traps for finite systems of cooperating compass automata // J. Inf. Process. Cybern. EIK 1987. Vol. 28, no. 8/9. Pp. 453–470.
- 6. Kilibarda G. On the minimum universal collectives of automata for plane labyrinths // Discrete Math. Appl. 1993. Vol. 3, no. 6. Pp. 555–586.
- 7. Szepietowski A. A finite 5-pebble-automaton can search every maze // Information Processing Letters. 1982. Vol. 15, no. 5. Pp. 199–204.
- 8. Килибарда Г. Новое доказательство теоремы Будаха-Подколзина // Дискретная математика. 1991. Т. 3, вып. 3. С. 135—146.
- 9. Килибарда Г., Ушчумлич Ш. О лабиринтах-ловушках для коллективов автоматов // Дискретная математика. 1993. Т. 5, вып. 2. С. 29–50.
- 10. Килибарда Г. О сложности автоматного обхода лабиринтов // Дискретная математика. 1993. Т. 5, вып. 3. С. 116–124.

Эксперименты с автоматами в алгебраически определенных классах

К.ф.-м.н. Козловский В. А.,

зав. лабораторией «Дискретной математики и прикладной алгебры» ИПММ НАН Украины, Украина, г. Донецк, ул. Р. Люксембург, 74, e-mail: kozlovskii@iamm.ac.donetsk.ua

Толмачевская Л. А.,

ИПММ НАН Украины, Украина, г.Донецк, ул.Р.Люксембург, 74, e-mail: lilja@pisem.net

Ввеление.

В работе рассматривается задача однозначного восстановления автомата A, принадлежащего заданному классу F, по информации о его поведении, полученной в результате эксперимента с ним [1]. Такой эксперимент обычно представляет собой некоторое множество вход-выходных слов, порожденных автоматом A, и в случае, когда по эксперименту однозначно определяется автомат A, его принято называть контрольным.

Основой для изучения свойств таких экспериментов являются свойства заданных автомата-эталона A и класса автоматов F .

Существуют различные способы задания классов: через указание параметров входящих в него автоматов (так называемые n-полные классы автоматов, состоящие из автоматов с n состояниями), заданием возможных автоматных отображений, описываемых недетерминированными автоматами, указанием стуктурно-алгебраических свойств входящих в класс автоматов и ряд других [2]. В [2] введено обобщающее различные способы задания автоматов (его спецификация) понятие дескриптора автоматов, и изучена связь свойств дескрипторов со свойствами контрольных экспериментов.

В настоящей работе в рамках исследования контрольных экспериментов предлагается новый способ описания классов автоматов, опирающийся на понятие так называемой определяющей пары автомата [3]. Основная идея такого описания состоит в расширении (аппроксимации) исходного класса F до некоторого более широкого класса F(R) автоматов, которые характеризуются наличием в них одного и того же множества R входных циклических слов и проведение контрольных экспериментов относительно класса F заменяется на таковое относительно класса F(R). Такое расширение позволяет vчесть при исследовании алгоритмических и сложностных характеристик контрольных экспериментов структурно-алгебраические свойства автоматов. При построении контрольных экспериментов относительно достаточно широких классов автоматов важными становятся особенности поведения автомата-эталона, позволяющие фиксировать в эксперименте некоторые "точки", соответствующие одним и тем же состояниям автомата. Фиксация таких "точек" является достаточным, а в ряде случаев и необходимым условием для того, чтобы эксперимент был контрольным, и для этого используются так называемые идентификаторы состояний [2] автомата – специальные вход-выходные слова, включаемые в эксперимент. Существуют различные способы проявления этих идентификаторов. Отвлекаясь от структуры идентификаторов и способов их проявления, приходим к понятию размеченного эксперимента, обобщающего понятие обычного эксперимента и расширяющего понятие контрольного эксперимента на случаи, когда обычный эксперимент отсутствует (например, в бесконечных классах). Некоторые результаты, касающиеся характеризаций таких экспериментов были анонсированы авторами

Основная цель данной работы — описать метод аппроксимации произвольных классов автоматов алгебраически определенными классами, предложить способы построения экспериментов относительно таких классов и оценить их метрические характеристики, используя результаты для размеченных экспериментов. В целях упрощения изложения рассмотрение этой задачи проводится для групповых автоматов. Помимо важности такого типа автоматов, введение такого свойства позволяет более ясно описать структуру эксперимента и саму идею метода.

1. Размеченные эксперименты.

Рассматриваются приведенные связные автоматы Мили $A = (S_A; X; Y; \delta_A; \lambda_A)$, где S_A – множество состояний, X и Y – входной и выходной алфавиты, S_A и S_A – функции переходов и выходов автомата

A . Автомат без выхода $\overline{A} = (S_A, X, \delta_A)$ называется полуавтоматом автомата A . Инициальный автомат с начальным состоянием s_0 обозначим (A, s_0) .

Каждому символу из X ставится в соответствие отображение множества состояний в себя $\delta_x:S \to S$, $\delta_x(s)=\delta(s,x)$. Множество всех таких отображений как образующих порождает полугруппу G(A) преобразований множества состояний, которая называется опорной полугруппой автомата A. Автомат называется групповым или перестановочным, если его опорная полугруппа является группой.

Входной алфавит расширяется введением обратных символов $\overline{X} = X \cup X^{-1}$, и функции автомата естественным образом продолжаются на расширенный алфавит.

Группе G = < G, > ставится в соответствие автомат $G' = (G, X, \Delta) \mid \Delta(g, x) = g \cdot \varphi(x)$, $g \in G$, $x \in X$ и φ - некоторое отображение множества входных символов в множество образующих.

Если $\lambda_A(s,p)=q$, $s\in S_A$, $p\in \overline{X}^*$, $q\in Y^*$, то пару w=(p,q) называем вход-выходным словом, порожденным автоматом A, а слово p — первой проекцией pr_1w слова w.

Пусть A - конечный связный приведенных групповой автомат, а F(X,Y) - класс всех связных приведенных групповых автоматов в алфавитах X и Y.

Обычно контрольным экспериментом с автоматом A относительно заданного класса автоматов F называется такое множество вход-выходных слов, порождаемых некоторым состоянием автомата, что его порождение некоторым состоянием другого автоматом B из заданного класса влечет изоморфизм автоматов A и B.

Однако, например, в случае класса F(X,Y) не существует контрольных экспериментов в классическом понимании. Введем новое понятие размеченного эксперимента, которое обобщает понятие обычного эксперимента.

Пусть α - символ, который будем называть меткой. Выберем некоторое состояние $s \in S_A$ и присвоим ему метку α . Тогда вход-выходному слову w=(p,q), где $p=p_0p_1p_2...p_{k+1}$ и $q=q_0q_1q_2...q_{k+1}$, порожденному состоянием $t \in S_A$, поставим в соответствие размеченное слово $w(\alpha,s)=(p_1,q_1)\alpha(p_2,q_2)\alpha...\alpha(p_k,q_k)\alpha(p_{k+1},q_{k+1})$ такое, что $\delta(t,p_0)=s$, $\lambda(t,p_0)=q_0$, $\delta(s,p_1)=\delta(s,p_1p_2)=...=\delta(s,p_1...p_k)=s$, где $\lambda(s,p_1)=q_1$, $\lambda(s,p_1p_2)=q_1q_2$, ..., $\lambda(s,p_1...p_k)=q_1...q_k$. Множество всех размеченных экспериментов автомата A обозначим через Φ_A . В более общем случае можно рассматривать размеченные эксперименты с несколькими метками, но в данном случае ограничваемся минимальными средствами выделения состояний.

Пусть F - некоторый класс автоматов и $A \in F$ - эталон. Размеченный эксперимент $W_{\alpha} \in \widetilde{\Phi}_A$ назовем контрольным размеченным экспериментом, если из порождения этого размеченного эксперимента другим автоматом $B \in F$ следует изоморфизм автоматов A и B.

С целью анализа размеченных экспериментов с каждым таким экспериментом W_{α} свяжем соответствующее ему циклическое сужение $C(W_{\alpha}) = (p_1p_2...p_k, q_1q_2...q_k)$, полученное отбрасыванием начального и заключительного отрезков размеченного слова, не содержащих метки. Это преобразование сохраняет свойство быть контрольным экспериментом.

Теорема 1 [4]. Размеченный эксперимент W_{α} является контрольным тогда и только тогда, когда его циклическое сужение $C(W_{\alpha})$ является контрольным размеченным экспериментом.

Эта теорема доказывается с помощью специального преобразования эксперимента W_{α} , названного групповым замыканием. С каждым экспериментом W_{α} связывается частичный автомат $\mathcal{O}(W_{\alpha})$ -фрагмент автомата [2] с отмеченными состояниями, и для него определяется групповое замыкание, подобное замыканию фрагмента по идентификаторам состояний [2].

Групповое замыкание $[\Phi(W_{\alpha})]_{\Gamma}$ – это частичный автомат, полученный из $\Phi(W_{\alpha})$ отождествлением отмеченных состояний и последующей детерминизацией полученного, вообще говоря, недетерминированного автомата в прямом и обратном направлении переходов.

Теорема 2. Размеченный эксперимент W_{α} автомата (A,s_0) является его контрольным размеченным экспериментом относительно класса F тогда и только тогда, когда его групповое замыкание $\left[\Phi(W_{\alpha})\right]_{\Gamma}$ является полностью определенным групповым гомоморфным прообразом автомата (A,s_0) .

2. Циклические эксперименты. С целью прояснения структуры размеченных экспериментов вводятся циклические эксперименты.

Циклическим относительно некоторого состояния $s\in S_A$ называется такое слово $p\in \overline{X}^*$, что $\delta_A(s,p)=s$. Если это свойство выполняется для всех $s\in S_A$, то $p\in \overline{X}^*$ называется циклическим словом автомата A.

По каждому размеченному эксперименту W_{α} строится множество $EC(W_{\alpha})$ его подслов $EC(W_{\alpha}) = \{w \,|\, W_{\alpha} = w' \alpha w \alpha w'' \wedge \alpha \not\in w\}$. Это множество называется циклическим экспериментом автомата A.

Каждому циклическому эксперименту W автомата A соответствует бесконечное множество размеченных экспериментов.

Контрольным циклическим экспериментом автомата A относительно класса F называется такой циклический эксперимент, который не является циклическим экспериментом никакого другого автомата этого класса.

Назовем два размеченных эксперимента эквивалентными, если множества построенных по ним циклических экспериментов, совпадают.

Замечание. Размеченный эксперимент W_{α} является контрольным для автомата (A,s_0) относительно класса F тогда и только тогда, когда $EC(W_{\alpha})$ - контрольный циклический эксперимент (A,s_0) относительно F .

Таким образом, класс эквивалентных размеченных экспериментов редуцируется множеством циклических экспериментов.

Для характеризации контрольных циклических экспериментов используются алгебраические свойства групповых автоматов.

Зададим автомат с помощью определяющей пары множеств циклических слов. Рассматриваются автоматы без выхода.

Пусть M=(R,L) – пара множеств слов в алфавите \overline{X} .

Если для некоторого слова $p \in R$ слово $v \in \overline{X}^*$ имеет вид $v = v_1 p v_2$, где $v_1, v_2 \in \overline{X}^*$, а слово $w = v_1 q v_2$, $q \in R \cup \{e\}$, то говорят, что v непосредственно выводимо из w относительно R. Слово v непосредственно выводимо из v либо относительно v, в котором каждое v, в котором каждое v, в котором каждое v, в котором каждое v, изоморфен автомату v, то пара множеств v наименьшая правая конгруэнция, порожденная на свободной группе v0 указанным отношением выводимости. Если фактор-автомат v0 науморфен автомату v0 на пара множеств v0 называется определяющей парой автомата v1 за за за за v2 называется определяющей парой автомата v3 на за v3 на за v4 на за за v5 на за ва операции группы, а v6 на за операции группы, а v7 на v8 на v9 на v9 на за операции группы, а v9 на v9 на v9 на v9 на за операции группы, а v9 на v

Следующий критерий описывает алгебраическую структуру определяющей пары M=(R,L) автомата.

Теорема 3 [4]. Пара M=(R,L) является определяющей для автомата A тогда и только тогда, когда группа $Q=\langle X,R \rangle$, заданная образующими X и определяющими словами R, содержит

такую подгруппу $J\subseteq Q$ со множеством порождающих L, что A изоморфен фактор-автомату Q'/J автомата Q', соответствующего группе Q, по конгруэнции, определенной на ней правыми смежными классами группы Q по подгруппе J.

3. Алгебраически определенные классы автоматов.

Пусть A - некоторый приведенный групповой автомат и $\Lambda_A = \{\lambda_s \mid s \in S_A\}$, где $\lambda_s : X^* \to Y^*$ - автоматное отображение, соответствующее состоянию $s \in S_A$, $\lambda_s(p) = \lambda(s,p)$. Через D(A) обозначим класс автоматов B таких, что $\delta_A = \delta_B$ и $\Lambda_A \neq \Lambda_B$. Будем рассматривать такие классы автоматов F, что $F = \bigcup_{A \in F} D(A)$. Такие классы назовем полными по выходам. Ясно, что если A -

приведенный и принадлежит такому классу, то контрольный эксперимент с A относительно класса F является обходом автомата A (по всем его дугам). Приведенным по выходам является, например, часто рассматриваемый класс так называемый n-полный класс всех автоматов с n состояниями или его подкласс групповых автоматов. Класс F назовем алгебраически определенным, если в нем существует такой автомат Γ , что для любого гомоморфного образа \overline{B} его полуавтомата $\overline{\Gamma}$ в классе F существует автомат B с полуавтоматом \overline{B} .

Через $\langle F \rangle$ обозначим наименьший по включению алгебраически определенный класс, содержащий класс F. Будем полагать, что F состоит из связных групповых автоматов и является полным по выходам. Так как класс F(X,Y) является алгебраически определенным, то для всякого его подкласса $F \subseteq F(X,Y)$ класс $\langle F \rangle$ существует.

На примере класса $F=G_N$ всех групповых автоматов с N состояниями рассмотрим аппроксимацию класса F алгебраически определенным классом $\langle F \rangle$ и связь между контрольными экспериментами связного приведенного автомата $A \in F$ относительно F и относительно $\langle F \rangle$. Для простоты будем рассматривать случай, когда A является определенно диагностируемым порядка 1 автоматом (ОД-1), то есть таким, у которого любое входное слово является диагностическим.

Для любого полного по выходам класса F и эталона $A \in F$ справедливо очевидное

Утверждение. Всякий контрольный эксперимент инициального автомата (A, s_0) относительно класса F является обходом A.

Обход понимается в широком смысле (не только для простых, но и для кратных экспериментов) как множество слов, по которым из начального состояния проходятся все дуги автомата.

Пусть w=(p,q) - эксперимент ОД-1-автомата $(A,s_0)\in G_N$, являющийся обходом. В этом случае достаточно просто строится размеченный эксперимент w_α . Состоянию s_0 ставим в соответствие метку α . Для любых p_1 , x, p_2 , если $p=p_1xp_2$, $q=q_1yq_2$, $\delta(s_0,p_1)=s_0$, $\lambda\left(s_0,p_1\right)=q_1$, то после p_1 ставится метка α . В результате получим размеченный эксперимент w_α . В силу того, что автомат A ОД-1 автомат с N состояниями, и w - обход, то для любого другого автомата $B\in G_N$, для которого w является экспериментом, w_α будет размеченным экспериментом B .

Так как $G_N \subseteq \langle G_N \rangle$, то всякий контрольный эксперимент (размеченный или нет) автомата A относительно $\langle G_N \rangle$ будет таковым относительно G_N . В [5] найдены необходимые и достаточные условия, при которых W является контрольным экспериментом группового ОД-1 автомата A относительно класса G_N . Основным условием при том является однозначная N -раскрашиваемость специального графа G(W), который с привлечение операции группового замыкания может быть построен следующим образом.

По эксперименту w (предполагаем, что w - обход A), строится фрагмент $\Phi(w)$ и выполняется его групповое замыкание $[\Phi(w)]_\Gamma$. Множество состояний полученного автомата является множеством вершин графа G(w). Для каждой вершины v выделяется подмножество $X(v) \subseteq X$ входов, при которых определена функция переходов автомата $[\Phi(w)]_\Gamma$, и ребро (v,v') принадлежит графу G(w), если $X(v) \cap X(v') \neq \emptyset$ или существует такая пара v'', v''' вершин, что для некоторого $x \in X(v'') \cap X(v''')$ есть переход из v'' в v и из v''' в v'.

Теорема 4. Если w есть контрольный эксперимент группового ОД-1 автомата A относительно класса G_N такой, что граф G(w) есть полный граф K_N , то w есть контрольный эксперимент A относительно класса $\langle G_N \rangle$.

С другой стороны, справедлива

Теорема 5 [4]. Множество P является контрольным циклическим экспериментом конечного связного группового автомата A относительно класса всех связных приведенных групповых автоматов тогда и только тогда, когда пара (\emptyset, pr_1P) является определяющей для связного конечного группового гомоморфного прообраза полуавтомата автомата A.

Из нее следует, что эксперимент для группового ОД-1 автомата, построенный на основе определяющей пары $M=(\varnothing,L)$ будет контрольным относительно класса G_N . Заметим, что при этом построение такого эксперимента осуществляется полиномиальным от параметров автомата алгоритмом [6].

В общем случае можно использовать специальные приемы формирования меток в размеченном эксперименте на основе различных определяющих пар, которые дают широкий спектр алгебраически определенных классов, аппроксимирующих заданный класс.

При этом метрические характеристики экспериментов могут быть использованы для оценок параметров экспериментов относительно базовых классов.

4. Метрические характеристики.

Для получения точных оценок кратности и суммарной длины использована описанная в [7] техника деревьев.

Кратность эксперимента – это число входящих в него слов, а длина – суммарная длина этих слов.

Параметрами являются n и m, где n – число состояний автомата, а n - мощность входного алфавита.

Обозначим d_{\min}^A - минимальная длина контрольного циклического эксперимента автомата A в ал ϕ авите \overline{X} .

$$d' = \frac{k}{2m-1} ((2m)^k (2m(2m-1)+1) - (2mn-n+1)) + (k+1)(2mn-n+1-(2m)^{k+1}) + mn-n+1 ,$$

где $k = [\log_{2m}(2mn - n + 1)] - 1$, $d'' = n \cdot (mn - n + 1)$.

Теорема 6. [7] Кратность приведенного контрольного циклического эксперимента P автомата B, такого, что $M=(\varnothing,pr_1P)$ — определяющая пара для некоторого A, равна mn-n+1, а для минимальной длины справедливы неравенства $d' \leq d_{\min}^A \leq d''$, причем обе оценки достижимы, а для любого числа $C=d''-c\cdot(2m-2)$, $(c\geq 0)$, $C\geq d'$, найдется групповой автомат A_i такой, что $d_{\min}^{A_i}=C$, а для любых других $C'\neq C$, d'< C'< d'', не существует автомата A такого, что $d_{\min}^A=C'$.

Заметим, что теорема 6 дает не только достижимые оценки, но и описывает спектр всех возможных значений функции минимальной длины.

Аналогичные оценки получены для контрольных экспериментов в алфавите X .

Обозначим $d_{\min}^{A,X}$ - минимальная длина контрольного циклического эксперимента автомата A в алфавите X

Теорема 7.[7] Для любого группового автомата A с число состояний n и мощностью входного алфавита m справедливы неравенства

$$2km^{k+1} - (2k+1)m + 2(k+1)m^2 + 1 - \frac{m^{k+1} - 1}{m-1} + mn - n + 1 \le d_{\min}^{A,X} \le \le n \cdot (mn - n + 1),$$

где $k = [\log_m(mn-n+1)]-1$, причем обе оценки достижимы.

Из теоремы 7 получается

Следствие. Для контрольных экспериментов групповых ОД-1 автоматов относительно класса G_N справедливо неравенство $d_{\min}^A \le n \cdot (mn - n + 1)$.

Заключение.

В работе описан метод аппроксимации произвольных классов автоматов алгебраически определенными классами. Для случая групповых автоматов предложены способы построения экспериментов относительно таких классов, использующие алгебраические свойства групповых автоматов. Приведены точные оценки кратности и минимальной длины таких экспериментов с описанием поведения функции длины.

Предложенный в работе подход к построению контрольных экспериментов на основе алгебраической аппроксимации классов может быть полезен при разработке полиномиальных алгоритмов построения контрольных экспериментов, получения оценок их параметров.

Список литературы

- 1. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов М.: Наука, 1985. 320 с.
- 2. Грунский И. С., Козловский В. А. Синтез и идентификация автоматов. Киев: Наукова думка, 2004. 246 с.
- 3. Толмачевская Л. А. Контрольные циклические эксперименты с групповыми автоматами // Кибернетика и системный анализ. 2005. № 3 С. 32 46.
- 4. В. А.Козловский, Л. А.Толмачевская Эквивалентность характеризаций экспериментов с групповыми автоматами // Седьмая Международная Конференция "Дискретные модели в теории управляющих систем" (Москва, 2006)
- 5. Козловский В. А. О представлениях групповых автоматов. // Кибернетика и системный анализ. 1996. № 2. C.21-28.
- 6. Толмачевская Л. А. Алгоритмы построения контрольных циклических экспериментов с групповыми автоматами // Труды института прикладной математики и механики.- 2006. Т.12.- С. 178-193
- 7. Толмачевская Л. А. Метод деревьев для получения оценок контрольных экспериментов с групповыми автоматами // VI Международная Конференция "Дискретные модели в теории управляющих систем" (Москва, 7-11 декабря 2004 г.). Москва, 2004. С. 144 147.

Идентификация типа среды в игровой постановке задачи о случайных блужданиях взаимодействующих частиц

Кольцов Д. А., Сердобольская М. Л.,

МГУ им. М.В. Ломоносова, Москва,

 $E ext{-}mail: damage@cmp.phys.msu.su, } maria@cmp.phys.msu.su$

Задачи о случайных блужданиях частиц, взаимодействующих между собой, рассматриваются в различных областях науки. Особенно детально и глубоко проработаны подходы кинетической теории [1,2,3,4] и теории ветвящихся случайных процессов с взаимодействием[5]. Однако существуют модели, анализ которых не сводится к решению задач, анализируемых в указанных теориях. Одна из таких моделей связана с задачей идентификации типа системы взаимодействующих частиц по наблюдениям за какой-либо выделенной частицей системы. При этом анализируемой системе свойственна неопределенность, содержащаяся в правилах перемещения и взаимодействия частиц. Именно эта неопределенность является основной проблемой задачи идентификации, т.к. при при заданных правилах перемещения и взаимодействия частиц процесс идентификации системы не вызывал бы особенных трудностей.

Рассмотрим следующую игровую модель двумерного дискретного случайного блуждания. Пусть N_0 взаимодействующих частиц совершают движение по квадратной сетке. Сетка разделена на одинаковые непересекающиеся области взаимодействия частиц. Динамика поведения и взаимодействия частиц описывается для дискретного времени. В начальный момент времени в заданных узлах сетки расположены частицы. В каждый последующий момент времени t(n)=n каждая частица смещается так, что двумерный радиус-вектор ее смещения задается формулой $\vec{\xi}(n)=\vec{X}(n)+\vec{\nu}(n)$, где $\vec{X}(n)$ – смещение, связанное с взаимодействием частиц, $\vec{\nu}(n)$ – случайная погрешность, $n=1,\ldots,T$. При этом используется следующая модель взаимодействия частиц:

- взаимодействуют только частицы, находящиеся в одной и той же области взаимодействия;
- с каждой частицей связан не зависящий от времени вектор \vec{R}_c , называемый *инструкцией*, $c = 0, 1, \dots, C$ номер частицы;
- каждая частица выбирает из инструкций, полученных от всех своих соседей (частиц, находящихся с ней в одной и той же области взаимодействия), наиболее типичную и совершает перемещение в соответствии с ней;
- наиболее типичная инструкция определяется как ближайшая к средней: для частиц с номерами c_1, \ldots, c_L

$$ec{R}_{c_*} = rg \min_{ec{R}_{c_i}} \left\| ec{R}_{c_i} - rac{1}{L} \sum_{j=1}^L ec{R}_{c_j}
ight\|$$

ullet если в момент времени t_n в одной области взаимодействия вместе с c-й частицей оказались L соседей, то вектор смещения c-й частицы есть

$$\vec{X}_c(t_n) = \left\{ \begin{array}{cc} \vec{X}_c(t_{n-1}), & L = 0 \\ \vec{R}_{c_*}, & L > 0 \end{array} \right. ;$$

ullet частица исполняет инструкцию с некоторой погрешностью, задаваемой с.в. $\vec{
u}$.

Будем считать, что наблюдению доступны перемещения одной фиксированной частицы, скажем, частицы с номером c=0. Остальные частицы влияют на ее движение, т.е. моделируют влияние среды, и их перемещения наблюдению недоступны. Для любой частицы ее инструкция $\vec{R}_c \in \mathcal{H}$, $c=1,\ldots,C$, где \mathcal{H} – некоторое конечное множество двумерных векторов. Мы расматриваем случай, когда каждая из координат любого вектора \vec{R}_c измеряется целым числом некоторых единиц (шагов сетки), каждая координата любого случайного вектора погрешности $\vec{\nu}(n)$ распределена дискретно, и ее значения также измеряются целым числом шагов сетки, $c=1,\ldots,C$; $n=1,\ldots,T$.

Предположим, что заданы два множества $\mathcal{H}^v \subset \mathbb{R}^2, v=1,2$, возможных инструкций, отвечающие двум разным типам сред. На основании реализаций $\vec{x}_0(n), n=1,\ldots,T$, смещений наблюдаемой частицы необходимо принять решение в пользу первого или второго вида среды. Мы рассматриваем систему из 50 частиц, которые блуждают по сетке, составлленной из 8×10 областей взаимодействия размером 10×10 шагов. Множества \mathcal{H}^v мы задавали как $\mathcal{H}^v=H^v\otimes H^v$, где $H^v=\{f_1^v,\ldots,f_{m_v}^v\}$, f_i^v – некоторые числовые значения, $m_1=2,\,m_2=3$.

Отметим особенности данной задачи, выделяющие ее в ряду других задач анализа систем взаимодействующих частиц. Так, в кинетической теории предполагается соблюдение характерных для исследуемой системы физических законов, накладывающих условия на возможные способы перемещения и взаимодействия частиц. Кроме того, считается, что силы, действующие на частицы и моделирующие их взаимодействие, эволюционируют явно заданным (детерминированным или стохастическим) образом. Макроскопические свойства исследуемой системы получаются как решения систем стохастических уравнений при переходе к большому числу частиц в системе. Все это не позволяет данную задачу идентификации типа среды рассматривать как какую-либо задачу кинетической теории: мы не рассматриваем систему из большого числа частиц, процесс изменения инструкций (аналогов силам в кинетике) не предполагается подчиняющимися какому-либо детерминированному или стохастическому закону, а является неявной функцией характера эволюции системы.

В теории ветвящихся случайных процессов со взаимодействием необходимо прежде всего указать класс, к которому принадлежат инфинитеземальные характеристики системы и после проверки определенных условий решить комплекс уравнений для определения стационарных или финальных вероятностей (вероятностей вырождения процесса в поглощающих состояниях). Только после этого можно дать ответ на следующий вопрос: возможно ли заменить модель движения всех частиц системы моделью, в которой присутствовала бы всего лишь одна наблюдаемая частица. При этом взаимодействие ее со средой определяется тем, что в область взаимодействия с заданной вероятностью помещаются другие частицы, действующие на наблюдаемую в соответствии с определенной инструкцией. При наших размерностях системы (50 частиц, 8×10 областей взаимодействия размером 10×10 шагов, по 2 и 3 простых гипотезы в каждом виде среды) стохастический анализ системы сопряжен с большими аналитическими и вычислительными трудностями. В связи с этим нам не удалось применить теорию ветвящихся случайных процессов со взаимодействием к решению нашей задачи идентификации типа среды. Заметим также, что при выбранном нами уровне шума не дала результатов и простейшая обработка случайного блуждания, например, визуальный анализ траектории, выборочный анализ или расчет диаграммы направленности набора актов блужданий.

Все координаты случайного вектора $\vec{\nu}(n), n=1,\ldots,T$, мы считаем независимыми случайными величинами, а их распределение заданным. Если записать набор $\vec{\xi}(n), n=1,\ldots,T$, покоординатно в виде $\xi_n=x_n+\nu_n, n=1,\ldots,N$ (N=2T), то математическое ожидание $E\xi_n\in H^v$, и мы можем решать нашу задачу об идентификации типа среды как задачу проверки статистических гипотез о распределении N-мерного случайного вектора ξ . Отметим следующую особенность данной задачи. В силу сложного характера взаимодействия частиц нет возможности определить, в какой последовательности чередуются конкретные значения $E\xi_n=f_{i_n}^v\in H^v$ при изменении $n=1,\ldots,N$. Это отличает нашу задачу от традиционных постановок задач сложных гипотез [?]: неизвестно, какова детерминированная или стохастическая закономерность появления простых гипотез в соответствующей серии распределений элементов выборки или, более того, неизвестно, существует ли вообще такая закономерность. В таких «нестационарных» условиях теряет силу статистическая устойчивость частоты события в ряду наблюдений и претерпевает изменения традиционная интерпретация частоты как вероятности события. Подобная задача в теории возможностей возникает в связи с проблемой эмпирического восстановления возможности и рассмотрена в [7].

Мы предлагаем следующий алгоритм решения задачи.

ШАГ 1. Априори (т.е. независимо от полученной в эксперименте реализации случайного вектора ξ) ищем решение следующей минимаксной задачи:

$$L_* = \min_{\pi^1, \pi^2} \max \left(\max_{\gamma^2} L(\pi^1 \mid H^2, \gamma^2), \max_{\gamma^1} L(\pi^2 \mid H^1, \gamma^1) \right), \tag{1}$$

где $L(\pi^v \mid H^{\tilde{v}}, \gamma^{\tilde{v}})$ – риск, сопровождающий принятие решения в пользу H^v , при условии, что верна $H^{\tilde{v}}, v, \tilde{v} = 1, 2$. При этом векторы π^v с координатами $\pi^v(x_s), s = 1, \ldots, S$, задают рандомизированную процедуру принятия решения по одному наблюдению: если в эксперименте получена реализация x_s (в нашей постановке множество реализаций случайной величины ξ_n конечно), то с вероятностью $\pi^1(x_s)$ принимается гипотеза H^1 , с вероятностью $\pi^2(x_s) = 1 - \pi^1(x_s)$ принимается гипотеза H^2 . Вектор γ^v с координатами $\gamma_i^v, i = 1, \ldots, m_v$, где m_v – количество простых гипотез в гипотезе H^v , описывает указанную выше «нестационарность» распределения: γ_i^v есть доля координат вектора ξ , для которых $E\xi_n = f_i^v$, т.е верна i-я простая гипотеза.

ШАГ 2. Вводим величину $\tau_N^v(x)$ такую, что $\tau_N^v(x) = N_v/N$, если количество тех элементов N-мерного вектора x реализации ξ , по которым мы приняли гипотезу H^v , равно N_v , $N_1 + N_2 = N$. Величина $\tau_N^v(x)$ определяет частоту принятия гипотезы H^v в серии наблюдений x_{s_1}, \ldots, x_{s_N} в результате рандомизированной процедуры, определяемой векторами π_*^1, π_*^2 .

ШАГ 3. Разбиваем отрезок [0,1] значений τ_N^v на два непересекающихся подмножества. Если $\tau_N^v(x)$ попадает в v-е множество, то принимается гипотеза H^v , v=1,2. В сущности речь идет о голосовании в пользу той или иной гипотезы, осуществляемого по результатам каждого наблюдения в серии x_{s_1},\ldots,x_{s_N} .

На основе предлагаемого алгоритма, выбрав надлежащим образом множества принятия гипотез, мы построили статистический критерий, обладающий свойством состоятельности, и связали оценки вероятностей ошибочных решений с количеством наблюдений. Работа алгоритма продемонстрирована в вычислительном эксперименте.

Список литературы

- 1. Е. М. Лифшиц, Л. П. Питаевский, Физическая кинетика. М.: Физматлит, 2002.
- 2 Н. Г. Ван Кампен, Стохастические процессы в физике и химии. М.: Высшая школа, 1990.
- 3. Р. Балеску, Равновесная и неравновесная статистическая механика: В 2 т. М.: Мир, 1978.
- 4. В.И. Кляцкин, Динамика стохастических систем. М.: Физматлит, 2003.
- 5. А. В. Калинкин, Марковские ветвящиеся процессы со взаимодействием // УМН, 2002, т.56, №2., С. 23-84.
 - 6. А. А. Боровков, Математическая статистика. М.: Наука, 1984.
- 7. Ю. П. Пытьев, Возможность как альтернатива вероятности. Математические и эмпирические основы, применения. М.: Физматлит, 2006.

Вопросы конечности дифференциальных стандартных базисов

М. В. Кондратьева,

c.~ н.~ c.~ механико-математического ф-та MГУ, kondra <math>m@shade.msu.ru

Е. В. Панкратьев

в. н. с. механико-математического ф-та $M\Gamma Y$, pankrat@shade.msu.ru

А. И. Зобнин

м. н. с. механико-математического ф-та $M\Gamma Y$, $al_zobnin@mail.ru$

Д. Трушин

механико-математический факультет $M\Gamma Y$, _ aggel@mail.ru

Рассматриваются критерии конечности дифференциальных стандартных базисов идеалов колец дифференциальных многочленов от одной переменной. Обсуждается решение в частных случаях задачи вхождения в дифференциальный идеал. Устанавливается связь между конечностью стандартного базиса и радикальностью идеалов, порожденных многочленом порядка не выше первого.

Ключевые слова: дифференциальная алгебра, базисы Гребнера.

Стандартные базисы дифференциальных идеалов являются аналогами базисов Гребнера в обычных полиномиальных идеалах. В дифференциальной алгебре [9,6,4] первые обобщения понятия базиса Гребнера полиномиального идеала появились в конце 1980-х годов независимо друг от друга в работах Дж. Карра-Ферро [1,2] («дифференциальные базисы Гребнера») и Ф. Оливье [8](«стандартные базисы»). В начале 1990-х появилась работа Э. Мансфилд [7], в который дифференциальным базисом Гребнера был назван совсем другой объект. Оказалось, что эквивалентные свойства, которыми обладают обычные базисы Гребнера, оказываются уже не эквивалентными в дифференциальной алгебре. Поэтому обобщения базисов Гребнера можно делать по-разному, сохраняя то или иное свойство. Базисы Оливье и Карра Ферро всегда порождают идеал и (дифференциально) редуцируют все элементы идеала (и только их) к нулю. Базисы Мансфилд не обязаны порождать идеал. Чтобы избежать путаницы, базисы Оливье и Карра Ферро будут называться дифференциальными стандартными базисами.

Кольцом обыкновенных дифференциальных многочленов $\mathcal{F}\{x\}$ называется кольцо обычных многочленов от счетного числа переменных $\mathcal{F}[x,x_1,x_2,..]$ с оператором дифференцирования δ : $\delta x_i = x_{i+1}$. Дифференциальные многочлены являются аналогами алгебраических дифференциальных уравнений, в которых независимая переменная x играет роль неизвестной функции. Пусть $F \subset R\{x\}$. Минимальный дифференциальный идеал, содержащий F, обозначается через [F]. Пусть

 \mathbb{M} — множество всех дифференциальных мономов кольца $\mathcal{F}\{x\}$. Допустимым упорядочением называется линейный порядок \prec на \mathbb{M} , удовлетворяющий следующим свойствам:

- 1. $M \prec N \implies MP \prec NP \quad \forall M, N, P \in \mathbb{M};$
- 2. $1 \prec P \quad \forall P \in \mathbb{M};$
- 3. $x_i \prec x_j \iff i < j$.

Доказано [10], что эти свойства гарантируют полную упорядоченность М.

Рассмотрим дифференциальный идеал I в $\mathcal{F}\{x\}$. Множество $G\subset I$ называется дифференциальным стандартным базисом идеала I относительно \prec , если ΘG является (бесконечным) алгебраическим базисом Гребнера идеала I, рассматриваемого в кольце $\mathcal{F}[x_0,x_1,\ldots]$. Многочлен f называется квазилинейным относительно \prec , если $f\in \mathcal{F}$ или $\deg \operatorname{Im}_{\prec} f=1$.

Допустимое упорядочение ≺ называется:

- δ -лексикографическим, если $\operatorname{Im}_{\prec} \delta M = \operatorname{Im}_{\operatorname{lex}} \delta M \ \forall M \neq 1$ (примеры: lex, deglex);
- δ -фиксированным, если $\forall f \notin \mathcal{F}$ найдутся такие моном M и индекс r, что начиная с некоторого порядка дифференцирования будет иметь место равенство $\lim_{\prec} \delta^n f = M x_{r+n}$;
- *согласованным с квазилинейностью*, если производная любого квазилинейного многочлена сама является квазилинейной.
- β -упорядочением, если $\operatorname{Im}_{\prec} \delta^k x^n = \operatorname{Im}_{\operatorname{degrevlex}} \delta^k x^n \ \forall k, n \geqslant 1$ (примеры: degrevlex, wtrevlex);

Теорема 1. Любое δ -лексикографическое упорядочение согласовано с квазилинейностью и является δ -фиксированным. Любое β -упорядочение согласовано с квазилинейностью. Обратные утверждения неверны.

В отличие от обычных базисов Гребнера, дифференциальные стандартные базисы часто оказываются бесконечными. Не так давно авторы получили красивые примеры конечных стандартных базисов, а также некоторые критерии их конечности [?, ?], что дало новый импульс к развитию этой темы.

Теорема 2. (критерий конечности дифференциального стандартного базиса) Пусть $\prec - \delta$ -фиксированное упорядочение, согласованное с квазилинейностью, а I — собственный дифференциальный идеал в $\mathcal{F}\{x\}$. Идеал I обладает конечным дифференциальным стандартным базисом G относительно \prec если и только если в I содержится \prec -квазилинейный многочлен.

Теорема 3. Если идеал обладает конечным дифференциальным стандартным базисом при какомто δ -лексикографическом упорядочении, то он обладает им и при чистом лексикографическом упорядочении.

Теорема 4. Для того, чтобы идеал I обладал конечным стандартным базисом при β -упорядочении \prec , достаточно, чтобы либо I содержал \prec -квазилинейный многочлен, либо $I=[f^p]$, где $f-\prec$ -квазилинейный многочлен.

Имеется гипотеза, что это условие является и необходимым.

Важно получить новые критерии конечности и рекурсивности дифференциальных стандартных базисов при самых общих допустимых упорядочениях мономов.

На теореме 2 основан предложенный авторами процесс пополнения [12], возвращающий (при согласованном с квазилинейностью δ -фиксированном упорядочении) редуцированный дифференциальный стандартный базис идеала, если он конечен. Он называется «Улучшенным процессом Оливье». В отличие от оригинального процесса Оливье, улучшенный процесс гарантированно останавливается, если идеал обладает конечным стандартным базисом.

Пример. При лексикографическом упорядочении редуцированные дифференциальные стандартные базисы идеалов $[y_1^n + y], n \geqslant 3$, состоят из четырех многочленов:

- $y_1^n + y_0$;
- $n y_0 y_2 y_1^2$;
- $n y_1^{n-2} y_2^2 + y_2 = y_2 (n y_1^{n-2} y_2 + 1);$
- $y_3 n(n-2)y_1^{n-3}y_2^3$.

Видно, что они содержат квазилинейный многочлен (последний в списке). Можно показать, что все эти идеалы являются радикальными. Для идеала $[y_1^2 + y]$ это уже не верно: он не радикальный и обладает лишь параметрическим дифференциальным стандартным базисом.

Нахождение стандартного базиса тесно связано с решением задачи вхождения (т. е. с проверкой принадлежности многочлена дифференциальному идеалу). Для произвольных дифференциальных идеалов эта задача алгоритмически неразрешима [3], а для конечно порожденных идеалов к настоящему времени она решена только в некоторых частных случаях. К ним относятся: случай радикальных идеалов, изобарических (т.е. однородных по весу) идеалов, а также идеалов, обладающих конечным или параметрическим стандартным базисом (см.[11]). Эта задача была решена авторами для некоторых новых классов идеалов (например, для дифференциальных идеалов, порожденных композицией многочленов с определенными свойствами). В частности, задачу вхождения для класса идеалов, порожденных композицией многочленов, удалось свести к другой, более простой с алгоритмической точки зрения задаче вхождения [13].

Опираясь на работу Колчина [5] и используя полученные критерии конечности дифференциальных стандартных базисов, авторы установили связь между конечностью лексикографического базиса и радикальностью дифференциального идеала, порожденного многочленом порядка не выше первого. Кроме того, в этом случае имеется более тесная связь с экспонентой дифференциального идеала. Для изучения этого случая вводится понятие идеала сепарант дифференциального идеала $I: S_I := \{S_h \mid h \in I, h \notin \mathcal{F}\} \cup \{0\}$, где $S_h = \frac{\partial h}{\partial y_n}$, $n = \operatorname{ord} h$, — сепаранта многочлена h.

Предложение 1.

- S_I является (обычным) идеалом в $\mathcal{F}[y,y_1,y_2,\ldots]$.
- ullet $S_I=1\iff I$ содержит лексикографически квазилинейный многочлен.
- $\forall f \in \mathcal{F}{y} \setminus \mathcal{F}$ выполнено $[f] + (S_f) \subset S_{[f]} \subset [f] : S_f^{\infty} + (S_f)$.

Зафиксируем δ -фиксированное и согласованное с квазилинейностью допустимое упорядочение \prec (например, лексикографическое).

Теорема 5. Для многочлена f нулевого порядка следующие условия эквивалентны:

- [f] обладает конечным дифф. стандартным базисом относительно \prec ;
- [f] содержит \prec -квазилинейный многочлен;
- [f] радикальный;
- f свободен от квадратов.

Пример. Для f = ay + b, где $a, b \in \mathcal{F}$, идеал [f] обладает конечным лексикографическим дифференциальным стандартным базисом из одного элемента $\{f\}$, в то время как $[y^2]$ конечным стандартным базисом не обладает.

Пусть теперь $f = \sum_{i=0}^d Q_i(y) y_1^i \in \mathcal{F}[y,y_1]$ — многочлен *первого* порядка. Обозначим через $S_f = \sum_{i=1}^d i Q_i(y) y_1^{i-1}$ сепаранту f.

Теорема 6. Идеал [f] обладает конечным дифференциальным стандартным базисом относительно \prec если выполнены условия

- $[f]: S_f^{\infty} + (S_f) = 1,$
- $Q_2 \in \sqrt{(Q_0, Q_1)}$,
- (Q_0, Q_1^2) свободен от квадратов.

Из работы Колчина [5] следует, что эти условия достаточны для того, чтобы идеал [f] был радикальным.

Пример. Пусть $f_{m,n} = (y_1 + 1)^m - c y^n$, $c \in \mathcal{F}$, $c \neq 0$. Тогда $[f_{m,n}]$ является радикальным и обладает конечным лексикографическим дифференциальным стандартным базисом тогда и только тогда, когда $m \nmid n$. В этом случае в идеале $[f_{m,n}]$ содержится квазилинейный многочлен порядка $[\frac{n}{m}] + 3$.

Мы выдвигаем гипотезу, что условия теоремы 6 являются и необходимыми для радикальности [f]. Мы доказали это в большинстве частных случаев. Однако для высших порядков и неглавных дифференциальных идеалов это перестает быть верным.

Список литературы

- 1. G. Carrà Ferro. Gröbner Bases and Differential Algebra. Lecture Notes in Computer Science, 356:129–140, 1989.
- 2. G. Carrà Ferro. Differential Gröbner bases in One Variable and in the Partial Case. *Mathematical and Computer Modelling*, 25(8–9):1–10, 1997.
- 3. G. Gallo, B. Mishra, F. Ollivier, Some Constructions in Rings of Differential Polynomials, Lecture Notes in Computer Science, vol. 539, 171–182, 1991.
- 4. E. Hubert. Notes on triangular sets and triangulation-decomposition algorithms. I: Polynomial systems. II. Differential Systems. Lecture Notes in Computer Science, vol. 2630, 2003.
 - 5. E. R. Kolchin. On the Exponents of Differential Ideals. Annals of Mathematics, 42:740–777, 1941.
 - 6. E. R. Kolchin. Differential Algebra and Algebraic Groups. Academic Press, New York, 1973.
 - 7. E. Mansfield: Differential Gröbner Bases, Ph.D. Thesis, Univ. of Sydney, 1991.
- 8. F. Ollivier. Standard Bases of Differential Ideals. Lecture Notes in Computer Science, 508:304–321, 1990.
 - 9. J. F. Ritt. Differential Algebra. New York, 1950.
- 10. A. Zobnin. Essential Properties of Admissible Orderings and Rankings. In *Contributions to General Algebra*, volume 14, pages 205–221, 2004.
- 11. A. Zobnin. On Testing the Membership to Differential Ideals. In *Proceedings of CASC-2004*, pages 485–496, 2004.
- 12. A. Zobnin. Admissible Orderings and Finiteness Criteria for Differential Standard Bases. In *Proceedings of ISSAC-2005*, pages 365–372. ACM Press, 2005.
- 13. А. И. Зобнин, М. В. Кондратьева. Задача принадлежности для дифференциальных идеалов, порожденных композицией многочленов. Принято к печати в журнале *Программирование*, 2006.

О существовании решения модели динамики возрастной структуры популяции

Корниенко Т. Я.,

разработчик ПО ООО "ФЛОТОХИМ" 192212 г.Санкт-Петербург ул. Белградская 28-1-57 $m/\phi(812)$ -3610774 E-mail: tkorn@mail.ru

1. Введение. Одной из важных экологических проблем является анализ и прогноз изменений одного из основных внутренних факторов развития популяции — возрастной структуры, т.е. изменений количества особей различных возрастов [1]. Необходимость учета возрастной структуры динамики обусловлена следующими двумя положениями. Во-первых, два естественных процесса, изменяющих состав популяции, — размножение и смертность существенно зависят от количества в популяции особей определенного возраста, и следовательно соотношение между группами особей различного возраста определяет как способность популяции к размножению, так и ее выживаемость

в будущем [2]. Во-вторых, рациональная эксплуатация популяции, например, сбор "урожая"в подавляющем большинстве случаев требует учета возрастной структуры популяции [3].

2. Постановка задачи. Рассмотрим нелинейную модель Гартина-Макками [4], заменив функции смертности μ и функции рождаемости β зависимость от размера популяции $P(\wp+v)$ на более сложные зависимости Ψ и Φ , учитывающие неравномерность вклада в рождаемость и смертность особей разных возрастов

$$X_{0}(\wp) = \begin{cases} \gamma(-\wp), & \wp \in [-A, 0], \\ B(\wp), & \wp \in (0, T], \end{cases}$$

$$X(\wp, v) = X_{0}(\wp) + \int_{v_{0}(\wp)}^{v} \mu(\wp, \sigma, X(\wp, \sigma), \Psi(\wp, \sigma),) d\sigma,$$

$$\Psi(\wp, v) = \int_{\wp+v-A}^{\wp+v} \psi(\wp, v, \kappa) X(\kappa, \wp + v - \kappa) d\kappa,$$

$$B(\wp) = \int_{\wp-A}^{\wp} \beta(\sigma, \wp - \sigma, X(\sigma, \wp - \sigma), \Phi(\sigma, \wp - \sigma)) d\sigma,$$

$$\Phi(\sigma, \wp - \sigma) = \int_{\wp-A}^{\wp} \phi(\sigma, \wp - \sigma, \kappa) X(\kappa, \wp - \kappa) d\kappa.$$

$$\Omega_{0} = \left\{ (\wp, v) \in \mathbb{R}^{2} \mid v \in [0, A], \wp \in [-v, T - v] \right\},$$

$$v_{0}(\wp) = \begin{cases} -\wp, & \wp \in [-A, 0], \\ 0, & \wp \in [0, T], \end{cases}$$

Функции γ , β , μ , ψ , ϕ - известные непрерывно-дифференцируемые функции своих переменных. Неизвестная функция $X(\wp, v)$ называется динамикой возрастной структуры.

Условия существования решения системы (2) с линенйыми относительно X функциями β и μ расмотрены в статье [5]. В работе [6] найдены условия сущестования решения системы (2) с квадратичной относительно X и линенйую относительно Ψ функцией μ . Практические задачи [7] используют более сложные функции.

Далее система уравнений (2) рассматривается как система относительно неизвестных функций X, B, Ψ, Ψ при $(\wp, \upsilon) \in \Omega_0$.

3. Теорема

Пусть функции μ , β — непрерывно-дифференцируемы по своим аргументам, константы m_x , m_ψ , b_x , b_ϕ ограничвают соответствующие частные производные функций μ и β ; функции ψ , ϕ ограничены константой s соответственно при $\sigma \in [0,A]$, $v \in [0,A]$, $v \in [-v,T-v]$, выполнены условия

$$Am_x < 1, \quad u \quad \frac{AC}{1 - m_x A} < 1, \tag{3}$$

где $C = \max\{s[1+m_{\psi}A], b_x[1+m_{\psi}A] + b_{\psi}[1+m_xA]\}$, тогда система (2)) имеет единственное пешение.

Доказательство. Возьмем непрерывные функции B и Ψ , и $\wp \in [-A,T]$, тогда $\mu(\wp, v, X(\wp, v), \Psi(\wp, v)) = \bar{\mu}(\wp, v, X), v \in [v_0(\wp), A]$ и задача Коши

$$\frac{\partial X}{\partial v}(\wp, v) = \bar{\mu}(\wp, v, X), \quad v \in [v_0(\wp), A],
X(\wp, v_0(\wp)) = X_0(\wp),$$
(4)

где

$$X_0(\wp) = \begin{cases} \gamma(-\wp), & r \in [-A, 0], \\ B(\wp), & r \in [0, T] \end{cases}$$

имеет единственное решение

$$X(\Psi, B)(\wp, \upsilon) = X_0(\wp) + \int_{\upsilon_0(\wp)}^{\upsilon} \bar{\mu}(\wp, \sigma, X) \, d\sigma, \quad \upsilon \in [\upsilon_0(\wp), A].$$

Подставим полученное решение $X(\Psi,B)$ в остальные уравнения системы (2)

$$\Psi(\wp, \upsilon) = \int_{0}^{A} \psi(\wp, \upsilon, \sigma) \ X(\Psi, B)(\wp - \sigma, \sigma) \, d\sigma,
B(\wp) = \int_{0}^{A} \beta \left(\wp, \sigma, X(\Psi, B)(\wp, \sigma), \Phi(\wp, \sigma)\right) \, d\sigma,
\Phi(\wp, \upsilon) = \int_{0}^{A} \phi(\wp, \upsilon, \sigma) \ X(\Psi, B)(\wp - \sigma, \sigma) \, d\sigma.$$
(5)

Покажем, что оператор $G(\Psi, B, \Phi)$, соответствующий правым частям системы (5) в пространстве вектор-функций (Ψ, B, Φ) с нормой

$$\|(\Psi, B, \Phi)\| = \max \left\{ \max_{(\wp, \upsilon) \in \Omega_0} \|\Psi(\wp, \upsilon)\|, \max_{\wp, \in [-A, T]} \|B(\wp)\|, \max_{(\wp, \upsilon) \in \Omega_0} \|\Psi(\wp, \upsilon)\| \right\}$$

является сжимающим.

Расмотрим две вектор-функции $(\Psi_1, B_1, \Phi_1), (\Psi_2, B_2, \Phi_2) \in \Omega_{\Psi} \times \Omega_B \times \Omega_{\Phi}$ и соответствующие решения залачи Коши (4) $X_1(\wp, v) = X(\Psi_1, B_1)(\wp, v), X_2(\wp, v) = X(\Psi_2, B_2)(\wp, v),$ тогда для любых $(\wp, v) \in \Omega_0$

$$|X_{1}(\wp, \upsilon) - X_{2}(\wp, \upsilon)| \leq |B_{1}(\wp) - B_{2}(\wp)| + \int_{0}^{A} |\mu(\wp, \sigma, X_{1}(\wp, \sigma), \Psi_{1}(\wp, \sigma)) - \mu(\wp, \sigma, X_{2}(\wp, \sigma), \Psi_{2}(\wp, \sigma))| d\sigma.$$

Следовательно, при всех $(\wp, \upsilon) \in \Omega_0$

$$\begin{split} |X_1(\wp,\upsilon) - X_2(\wp,\upsilon)| &\leqslant \max_{\wp \in [-A,T]} |B_1(\wp) - B_2(\wp)| + \\ + Am_x \max_{(\wp,\upsilon) \in \Omega_0} |X_1(\wp,\sigma) - X_2(\wp,\sigma)| + Am_\Psi \max_{(\wp,\upsilon) \in \Omega_0} |\Psi_1(\wp,\sigma) - \Psi_2(\wp,\sigma)| \,. \end{split}$$

В силу условия $Am_x < 1$ имеет место оценка

$$\max_{(\wp,\upsilon)\in\Omega_0} |X_1(\wp,\sigma) - X_2(\wp,\sigma)| \leq \frac{1 + m_{\psi}A}{1 - m_{x}A} \|(\Psi_1, B_1, \Phi_1) - (\Psi_2, B_2, \Phi_2)\|.$$
 (6)

Оценим разность элементов вектор функции $G = (G_1, G_2, G_3)$

$$\begin{split} |G_{1}(\Psi_{1},B_{1},\Phi_{1})-G_{1}(\Psi_{2},B_{2},\Phi_{2})| &\leqslant \psi_{0} \int\limits_{0}^{A} |X_{1}(\wp,\sigma)-X_{2}(\wp,\sigma)| \ d\sigma, \\ |G_{2}(\Psi_{1},B_{1},\Phi_{1})-G_{2}(\Psi_{2},B_{2},\Phi_{2})| &\leqslant \phi_{0} \int\limits_{0}^{A} |X_{1}(\wp,\sigma)-X_{2}(\wp,\sigma)| \ d\sigma, \\ |G_{3}(\Psi_{1},B_{1},\Phi_{1})-G_{3}(\Psi_{2},B_{2},\Phi_{2})| &\leqslant b_{x} \int\limits_{0}^{A} |X_{1}(\wp,\sigma)-X_{2}(\wp,\sigma)| \ d\sigma + b_{\phi} \int\limits_{0}^{A} |\Phi_{1}(\wp,\sigma)-\Phi_{2}(\wp,\sigma)| \ d\sigma. \end{split}$$

Отсюда

$$||G(\Psi_1, B_1, \Phi_1) - G(\Psi_2, B_2, \Phi_2)|| \le \frac{AC}{1 - m_x A} ||(\Psi_1, B_1, \Phi_1) - (\Psi_2, B_2, \Phi_2)||.$$

В силу условия (3) оператор G — сжимающий. Теперь покажем полоноту пространства вектор-функций $Y=(X,\Psi,B,\Phi)$ с нормой

$$||Y|| = \max_{k} \max_{(\wp, \upsilon) \in \Omega_0} |Y_k(\wp, \upsilon)|.$$

Возьмем из этого пространства фундаментальную последовательность непрерывных вектор-функций $\{Y^{(i)}(\wp,v)\}$. Отсюда для любого $\varepsilon>0$ существует индекс i_0 такой, что для любых двух индексов $i_1,\,i_2>i_0$ выполнено

$$\max_{k} \max_{(\wp,\upsilon) \in \Omega_0} \left| Y_k^{(i_1)}(\wp,\upsilon) - Y_k^{(i_2)}(\wp,\upsilon) \right| < \varepsilon,$$

следовательно для всех элементов вектор-функции Y и любой точки $(\wp, \upsilon) \in \Omega_0$ выполняется условие Коши

$$\left| Y_k^{(i_1)}(\wp, \upsilon) - Y_k^{(i_2)}(\wp, \upsilon) \right| < \varepsilon$$

равномерной сходимости последовательности функций $\left\{Y_k^{(i)}(\wp,\upsilon)\right\}$. Таким образом, для каждого k последовательность $\left\{Y_k^{(i)}(\wp,\upsilon)\right\}$ сходится к непрерывной функции Y_k . Тогда

$$\lim_{k_2 \to \infty} \left| Y_k^{(i_1)}(\wp, \upsilon) - Y_k^{(i_2)}(\wp, \upsilon) \right| = \left| Y_k^{(i_1)}(\wp, \upsilon) - Y_k(\wp, \upsilon) \right| \leqslant 0, 5\varepsilon < \varepsilon$$

при всех k и любых $(\wp, v) \in \Omega_0$. Что означает

$$\max_{k} \max_{(\wp,\upsilon) \in \Omega_0} \left| Y_k^{(i_1)}(\wp,\upsilon) - Y_k(\wp,\upsilon) \right| < \varepsilon,$$

т.е. последовательность $\{Y^{(i)}(\wp, \upsilon)\}$ сходится к нерперывной вектор-функции $Y=(Y_1, \ldots, Y_4)$.

Из полноты пространства (X,Ψ,B,Φ) и сжимаемости оператора G вытекает существование единственной неподвижной точки. \square

Заключение. Сформулированы условия существования и единственности решения модели динамики возрастной структуры для непрерывно-дифференцируемых функций рождаемости м смертности.

Список литературы

- 1. *Елизаров Е. Я.*, *Свирижев Ю. М.* Математическое моделирование биологических систем// Проблемы космической биологии. М. 1972. No.20. C.3-7.
- 2. Popa V. T., Vlad O. M. A new nonlinear model of age-dependent population growth // Math. Biosc. 1985. Vol.76. P.161–184.
- 3. Swick K. E. A model of single species population dynamics // SIAM J.Math.Anal. 1977. Vol.32. P.484–498.
- 4. Gurtin M. E., MacCamy R. C. Nonlinear age-dependent population dynamics // Archs ration. Mech. Analysis. 1974. Vol.54, P.281–300.
- 5. Li J. Persistence and extinction in continuous age-structured population models// Comput. Math. Applic. 1988. Vol.15, No.6. P.511–523.
- 6. $\mathit{Mauehko}$ В. $\mathit{\Gamma}$. Анализ некоторых непрерывных моделей динамики возрастной структуры популяций. М., 1981. 72 с.
- 7. *Карелин В. В.*, *Корниенко Т. Я.* Оценка некоторых параметров дифференциальных уравнений с частными производными // Математические методы исследования управляемых динамических систем. СПб. 2000. С.77–82.

Параллельные алгоритмы поиска решений оптимизационных задач на масштабно-инвариантных графах большой размерности ¹

Кочкаров А. А.,

заместитель директора НОЦ "Прикладная математика" Института прикладной математики им. М.В. Келдыша РАН 125047, Москва, Миусская пл., 4 E-mail: Azret Kochkarov@mail.ru

Кочкаров Р. А.

старший преподаватель Финансовой Академии при Правительстве РФ 125468, Москва, Ленинградский пр., 49 E-mail: Rasul Kochkarov@mail.ru

Современные исследования сложных систем таких, как информационные, электроэнергетические, WWW (Internet), нейронные сети, метаболические сети клеток, сети научного сотрудничества показывают, что структуры этих систем по истечении времени претерпевает определенные изменения, вызываемые различными внешними обстоятельствами. Структуру системы, произвольной природы (социальной, социально-экономической, технической, химико-биологической и т.п.) можно представить в виде графа. Граф [1] — это абстрактный объект, как правило, вершины графа соответствуют элементам системы, а ребра — связям между элементами этой системы. Изменения, происходящие в структуре сложной системы, могут быть описаны простейшими теоретико-графовыми операциями [1]: стягивание ребра, удаление (добавление) ребра, удаление (добавление) вершины. Изменения структуры системы могут быть разовыми, а могут быть постоянными. Для второго случая, разумно, ввести понятие структурной динамики — изменение структуры системы с течением времени. Несомненно, для описании структурной динамики лучше всего подходит аппарат теории графов.

Одним из наиболее распространенных сценариев структурной динамики является *рост структуры*. Рост структуры – это регулярное появление новых элементов и связей в структуре системы. Рост структуры может происходит по строго сформулированным правилам, не исключая наличие в них фактора случайности.

 $^{^{1}}$ Работа выполнена при поддержке РФФИ (проект № 04-01-00510) и РГНФ (проект № 05-03-03188).

Исследование структурной динамики, как модели изменчивости связей информационных сетей и систем, представляется важной актуальной задачей. Изменение структуры информационных систем (сетей), вызванное выходом из строя элементов этой системы (сети), некоторым негативным образом отразится на ее качественных и количественных характеристиках. Положение некоторых элементов в структуре информационной (коммуникационной) сети могут оказаться более значимыми чем у остальных, поскольку выход из строя таких элементов в состоянии существенно ухудшить функционирование всей сети. При росте информационной сети важно не допускать появление таких связей и элементов, и целых подструктур (набор взаимодействующих элементов системы (сети)). Это задача усложняется тем, что с одной стороны рост – динамическое развитие можно наблюдать во многих "местах" сети одновременно. С другой стороны, нелегко распознать правила такого роста – "время и место" появления новых элементов и связей в сети. Вообще говоря, возможны различные правила структурной динамики.

В настоящей работе рассматривается одно из возможных правил задающих структурную динамику сложных информационных сетей. Формальным представлением изменения структур информационных сетей по этому правилу являются масштабно-инвариантные или самоподобные [2] графы большой размерности, называемые фрактальными (предфрактальные).

Понятие фрактал, введенное Бенуа Мандельбротом, объединило объекты обладающих особым свойством – свойством самоподобия (self-similarity) или масштабной инвариантности. Работы связанные с исследование фрактальных объектов (фрактальных множеств) долгое время считались занимательными, но неимеющие значительных приложений. Мнения в мировой научной среде изменились с изданием книги [3]. В настоящее время о перспективности и значимости исследований связанных с фрактальными множествами можно судить по регулярно проводимым конференциям и периодическим изданиям (см., например, журнал "Chaos, Solitons & Fractals"), целиком посвященных соответствующей тематике. Это позволяет говорить об сформировавшемся круге прикладных физических модельных задач на основе фрактальных множеств [3]. Среди них выделяются задачи и модели, где фрактальные множества представлены как самоподобные (фрактальные или масштабно-инвариантные) графы большой размерности, т.е. с большим количеством вершин. К ним относятся, например, задачи о броуновском движении (случайном блуждании), диффузии и просачиваемости. Кроме того, самоподобные графы нередко выступаю в качестве моделей структур сложных многоэлементных систем [4] таких, как коммуникационные сети. И в первом, и во втором случае возникают оптимизационные задачи, решение которых требует значительных временных затрат. А это, в свою очередь, приводит к необходимости использования методов параллельных вычислений. К сожалению, работы в этом направлении не многочисленны [5,6].

В настоящей работе предлагается описание *параллельного алгоритма* \Re поиска остовного дерева минимального веса (ОДМВ) на предфрактальном графе – конечном аналоге фрактального графа. Прежде чем перейти к описанию алгоритма, введем понятие фрактального (предфрактального) графа.

Термином затравка условимся называть какой-либо связный граф H=(W,Q). Суть операции ЗВЗ заключается в следующем. В данном графе G=(V,E) у намеченной для замещения вершины $\tilde{v}\in V$ выделяется множество $\tilde{V}=\{\tilde{v}_j\},\ j=1,2,...,\left|\tilde{V}\right|,$ смежных ей вершин. Далее из графа G удаляется вершина \tilde{v} и все инцидентные ей ребра. Затем каждая вершина $\tilde{v}_j\in \tilde{V},\ j=1,2,...,\left|\tilde{V}\right|,$ соединяется ребром с одной из вершин затравки H=(W,Q). Вершины соединяются произвольно (случайным образом) или по определенному правилу, при необходимости.

Предфрактальный граф будем обозначать через $G_L = (V_L, E_L)$, где V_L – множество вершин графа, а E_L – множество его ребер. Определим его рекуррентно, поэтапно заменяя каждый раз в построенном на предыдущем этапе l=1,2,...,L-1 графе $G_l=(V_l,E_l)$ каждую его вершину затравкой H. На этапе l=1 предфрактальному графу соответствует затравка $G_1=H$. Об описанном процессе говорят, что предфрактальный граф G_L порожден затравкой H. Процесс порождения предфрактального графа G_L по существу есть процесс построения последовательности предфрактальных графов $G_1, G_2, ..., G_l, ..., G_L$, называемой траекторией. Фрактальный граф G_L порожденный затравкой H, определяется бесконечной траекторией. Предфрактальный граф G_L условимся называть (n,q,L)-графом, если он порожден n-вершинной q-реберной связной затравкой H.

Для предфрактального графа G_L ребра, появившиеся на l-ом, $l \in \{1, 2, ..., L\}$, этапе порождения, будем называть ребрами ранга l. Новыми ребрами предфрактального графа G_L назовем ребра ранга L, а все остальные ребра назовем cmapumu.

При удалении из предфрактального графа G_L всех ребер рангов l=1,2,...,L-r получим множество $\{B_{L,i}^{(r)}\},\ r\in\{1,2,...,L-1\},$ блоков r-го ранга, где $i=1,2,...,n^{L-r}$ порядковый номер блока. Термином nodграф-затравка $z_s^{(l)}$ будем называть блок $B_{l,s}^{(1)},\ s=\overline{1,n^{l-1}}$ первого ранга

предфрактального графа G_l , $l=\overline{1,L}$ из траектории. Мощность множества $Z(G_L)=\{z_s^{(l)}\},\ l=\overline{1,L},$ $s=\overline{1,n^{l-1}}$ всех подграф-затравок из траектории графа G_L равна $|Z(G_L)|=\frac{n^L-1}{n-1}$.

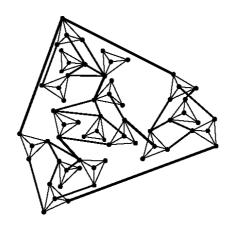


Рис. 1: $G_3 = (V_3, E_3)$

На рис. 1 изображен предфрактальный графа $G_3=(V_3,E_3)$, порожденного затравкой H – полным 4-вершинным графом. Самыми "жирными" линиями нарисованы ребра подграф-затравки $z_1^{(1)}$. Линиями средней "жирности" нарисованы ребра подграф-затравок $z_1^{(2)}$, $z_2^{(2)}$, $z_3^{(2)}$ и $z_4^{(2)}$. И наконец, тонкими линиями нарисованы новые ребра предфрактального графа G_3 , которые образуют подграф-затравки $z_s^{(3)}$, $s=\overline{1,16}$.

Будем говорить, что предфрактальный граф G_L взвешен, если каждому его ребру $e^{(l)} \in E_L$ приписано действительное число $w(e^{(l)}) \in (\theta^{l-1}a,\theta^{l-1}b)$, где $l=\overline{1,L}$ – ранг ребра, a>0, и $\theta<\frac{a}{b}$.

Алгоритм \Re осуществляет поиск ОДМВ $T=(V_L,E_T)$ на взвешенном предфрактальном графе G_L . Алгоритм использует k процессоров $p_1,p_2,...,p_k$ на многопроцессорной вычислительной машине с распределенной памятью.

Назначим каждый процессор одной из подграф-затравок $z_s^{(l)}$, $l=\overline{1,L}$, $s=\overline{1,n^{l-1}}$ предфрактального графа G_L , тогда число используемых процессоров равно $k=\frac{n^L-1}{n-1}$. Суть работы алгоритма заключается в следующем. Каждая подграф-затравка $z_s^{(l)}$ рассматривается как отдельно взятый граф. При этом каждый из k процессоров p_i , $i=\overline{1,k}$ параллельно независимо друг от друга находит ОДМВ T_i на своей подграф-затравке $z_s^{(l)}$. Поиск ОДМВ отдельно взятой подграф-затравки осуществляется алгоритмом Прима. Алгоритм Прима используется в алгоритме β_2 в виде процедуры по мере необходимости. Нахождение ОДМВ $T_1, T_2, ..., T_k$ всех подграф-затравок $z_s^{(l)}$ определяет ОДМВ T предфрактального графа G_L . Каждое ребро предфрактального графа имеет свой "уникальный" номер, однозначно определяющий ребро во всей траектории.

Таким образом, выделение ОДМВ на подграф-затравке $z_s^{(l)}$ соответствует выделению множества ребер на предфрактальном графе G_L . \lhd

Кроме принципиальной возможности эффективного распараллеливая задачи о поиске ОДМВ на предфрактальном графе, важен еще и следующий факт.

Теорема 1. Вычислительная сложность алгоритма \Re для предфрактального (n,q,L)-графа G_L , с числом вершин $|V_L|=N$, равна $O(N\cdot n^2)$.

Вычислительная сложность алгоритма Прима равна $O(N^2)$. Сравнив ее с вычислительной сложностью алгоритма \Re , получаем: $O(N \cdot n^2) < O(N^2)$. Это говорит о том, что при реализации алгоритма \Re на одном процессоре поиск ОДМВ на предфрактальном графе будет осуществлен быстрее, чем широко известным алгоритмом Прима.

В настоящей работе на примере параллельного алгоритма \Re поиска остовного дерева минимального веса на предфрактальном графе продемонстрирована идея распараллеливания оптимизационных задач на масштабно-инвариантных графах большой размерности.

Список литературы

1. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990.

- 2. Ахромеева Т.С., Курдюмов С.П., Малинецкий Г.Г., Самарский А.А. Нестационарные структуры и диффузионный хаос. М.: Наука, 1992.
 - 3. Фракталы в физике / Под ред. Л. Пьетронеро, Э. Тозатти. М.: Мир, 1988.
 - 4. Song C., Havlin S., Makse H.A. Self-similarity of Complex Networks // Nature 433, 392–395 (2005).
- 5. Riehl J., Hespanha J.P. Fractal graph optimization algorithms // Proc. of the 44th Conf. on Decision and Contr., 2005. P. 2188–2193.
- 6. Кочкаров А.А., Кочкаров Р.А. Параллельный алгоритм поиска кратчайшего пути на предфрактальном графе // Журнал вычисл. матем. и матем. физики. 2004. Т. 44, № 6. С. 1157–1162

Исследование полиномиальных алгоритмов решения четырехиндексной аксиальной проблемы выбора

Кравцов М. К.,

заведующий отделом экономико-математического моделирования Научно-исследовательского экономического института Министерства экономики Республики Беларусь доктор физико-математических наук, профессор,

Дичковская С. А.,

инженер-программист РУП "БелФармация" 220086 Минск, ул. Славинского, 1, корп.1, НИЭИ Мин. экономики РБ E-mail: dichkovskaya@mail.ru

Известно [1], что р–индексная $(p \geq 3)$ аксиальная проблема выбора (задача о назначениях), имеющая многочисленные практические применения (см., например, [2,3]), является NP-трудной. Для решения этой задачи в [4] разработаны и обоснованы р новых алгоритмов $\alpha_0, \alpha_1, \alpha_2, ..., \alpha_{p-1}$ нахождения асимптотически оптимального решения, использующих общеизвестную схему "жадных"эвристик, трудоемкость которых составляет соответственно $O(n^{p+1}), O(n^p), O(n^{p-1}), ..., O(n^2)$ операций. В [5] проведено вычислительное исследование алгоритмов α_0, α_1 и α_2 для 3–индексной аксиальной проблемы выбора.

В настоящей работе проведена серия вычислительных экспериментов по исследованию алгоритмов $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ применительно к 4-индексной аксиальной проблеме выбора (4-аксиальной ПВ) порядка n, постановка которой заключается в минимизации целевой функции

$$f(X) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \sum_{i_3=1}^{n} \sum_{i_4=1}^{n} c_{i_1 i_2 i_3 i_4} x_{i_1 i_2 i_3 i_4}$$

при условиях

$$\begin{split} & \sum_{i_1=1}^n \sum_{i_2=1}^n \sum_{i_3=1}^n x_{i_1 i_2 i_3 i_4} = 1 \quad \forall \, i_4 \in N_n, \qquad \sum_{i_1=1}^n \sum_{i_2=1}^n \sum_{i_4=1}^n x_{i_1 i_2 i_3 i_4} = 1 \quad \forall \, i_3 \in N_n, \\ & \sum_{i_1=1}^n \sum_{i_3=1}^n \sum_{i_4=1}^n x_{i_1 i_2 i_3 i_4} = 1 \quad \forall \, i_2 \in N_n, \qquad \sum_{i_2=1}^n \sum_{i_3=1}^n \sum_{i_4=1}^n x_{i_1 i_2 i_3 i_4} = 1 \quad \forall \, i_1 \in N_n, \end{split}$$

$$x_{i_1 i_2 i_3 i_4} = 0$$
 или $1 \quad \forall (i_1, i_2, i_3, i_4) \in N_n^4$,

где $C=\|c_{i_1i_2i_3i_4}\|_n$ — заданная 4-индексная матрица с действительными элементами, $N_n=\{1,2,\ldots,n\},\,N_n^4=N_n\times N_n\times N_n\times N_n,\,X=\|x_{i_1i_2i_3i_4}\|_n$ — исходный план 4-аксиальной ПВ.

Асимптотически оптимальные алгоритмы.

Зафиксируем некоторое число $q \in \{1,2,3\}$. Под (4-q)-мерным сечением 4-индексной матрицы $C = \|c_{i_1 i_2 i_3 i_4}\|_n$ с фиксированными значениями q индексов, например $i_1^0, i_2^0, ..., i_q^0$, будем понимать (4-q)-мерную подматрицу $C^{i_1^0 i_2^0 ... i_q^0} = \|c_{i_q + 1 i_q + 2... i_q}^{i_1^0 i_2^0 ... i_q^0}\|_n$, элементы которой определяются следующим образом

$$c_{i_{q+1}i_{q+2}\dots i_4}^{i_1^0i_2^0\dots i_q^0} = c_{i_1^0i_2^0\dots i_q^0i_{q+1}i_{q+2}\dots i_4} \quad \forall (i_{q+1},i_{q+2},...,i_4) \in N_n^{4-q}.$$

Естественно считать, что 4-мерное сечение 4-индексной матрицы C представляет собой эту же матрицу.

Зафиксируем $q \in \{0,1,2,3\}$. Опишем приближенный алгоритм α_q построения плана $X^{(q)} =$

 $\|x_{i_1i_2i_3i_4}^{(q)}\|_n$ 4-аксиальной ПВ. Он состоит из последовательно проводимых шагов. Шаг 1. Найти минимальный элемент в (4-q)-мерном сечении 4-индексной матрицы C с фиксированными значениями индексов $i_1^0, i_2^0, ..., i_q^0$. Если таким элементом оказался $c_{i_1^0i_2^0i_3^0i_4^0}$, то положить $x_{i_1^0 i_2^0 i_3^0 i_4^0}^{(q)} = 1$, $x_{i_1 \dots i_{t-1} i_t^0 i_{t+1} \dots i_4}^{(q)} = 0 \ \forall i_s \in N_n, \ \forall s \in N_4 \setminus \{t\}, \ \forall t \in N_4$. Далее, вычеркнув из матрицы C все 3-мерные сечения, содержащие элемент $c_{i_1^0 i_2^0 i_3^0 i_4^0}$, получим 4-индексную матрицу $C^1 = \|x_{i_1 i_2 i_3 i_4}^{(1)}\|_{n-1}.$

Ясно, что указанная процедура может быть продолжена для матрицы C^1 . Всего необходимо проделать n шагов этого процесса. В результате получаем некоторый план $X^{(q)}$ 4-аксиальной ПВ. Легко видеть, что трудоемкость алгоритма α_q , где $q \in \{0,1,2,3\}$, составляет $O(n^{5-q})$ операций.

Пусть V(n,r) – множество всех 4-индексных матриц $C = \|c_{i_1 i_2 i_3 i_4}\|_n$, элементы которых равновероятно и независимо принимают целочисленные значения из отрезка [1,r], где r=r(n) – либо константа, либо $r(n) \to \infty$ при $n \to \infty$. Пусть $V^{\sigma}(n,r)$ – множество всех тех матриц С из множества V(n,r), которые обладают свойством σ . Будем говорить, что почти каждая матрица множества V(n,r) обладает свойством σ , если $\lim_{n\to\infty} \frac{|V^{\sigma}(n,r)|}{|V(n,r)|} = 1$. Будем также говорить, что алгоритм α_q , где $q \in \{0, 1, 2, 3\}$, почти всегда строит асимптотически оптимальный план $X^{(q)}$ 4-аксиальной ПВ, если существует такая последовательность $\varepsilon_n(q) \geq 0$, $\lim_{n \to \infty} \varepsilon_n(q) = 0$, что для почти каждой матрицы C из множества V(n,r) выполняется неравенство $f(X^{(q)}) \leq f(X^0)(1+\varepsilon_n(q)),$ где X^0 оптимальный план 4-аксиальной ПВ.

Теорема (см. [4]). Для любого числа $q \in \{0,1,2,3\}$ алгоритм α_q почти всегда строит асимптотически оптимальный план 4-аксиальной ПВ, если $r \leq n^{\frac{1-\theta}{1+q}}$, где θ – любое число, удовлетворяющее неравенствам $1/5 < \theta < 1$ при $q=0; 0 \le \theta < 1$ при $q \in N_3$.

Вычислительное исследование асимптотически оптимальных алгоритмов.

Изложим результаты вычислительных экспериментов по исследованию алгоритмов α_0 , α_1 , α_2 и α_3 на тестовых 4-аксиальных ПВ. Тестовые задачи формировались следующим образом: 4индексная матрица С формировалась с помощью датчика случайных чисел, настроенного на работу с целыми числами из отрезка [1,r], где $2 \le r \le n$. Проверка плана X^* , построенного с помощью алгоритмов α_0 , α_1 , α_2 и α_3 , на оптимальность осуществлялась на основе следующего очевидного утверждения: если элементы матрицы С принимают целочисленные значения из отрезка [1,г], то всякий план X 4-аксиальной ΠB , для которого выполняется равенство f(X)=n, является оптимальным. Алгоритмы α_0 , α_1 , α_2 и α_3 программно реализованы на языке Object Pascal (в среде Delphi). По ним проведены вычислительные эксперименты с использованием трех двухпроцессорных серверов с Pentium 4, 3,2 ГГц, 2Гб. Для каждого алгоритма было решено 40000 задач (по 1000 задач для фиксированных $n \in \{50, 75, 100, 125, 150\}$ и $r \in \{2, \lceil \sqrt[3]{n} \rceil, \lceil \sqrt[4]{n^3} \rceil, \lceil \sqrt[5]{n^4} \rceil, \lceil n/2 \rceil, \lceil 3n/4 \rceil, n\}$). Здесь [m] – наибольшее целое число, не превосходящее числа m.

Результаты вычислительных экспериментов, проведенных на тестовых 4-аксиальных ПВ для алгоритмов α_0 , α_1 , α_2 и α_3 , представлены в табл. 1 и 2.

Таблица 1

		До	ля зада	ач,		Доля задач,				
	решаемых оптимально					решаемых оптимально				
	с помощью алгоритма					с помощью алгоритма				
	α ₀ (в %)					α_1 (B %)				
r/n	50	75	100	125	150	50	75	100	125	150
2	96.1	96.9	97.6	98.1	98.5	90.1	91.4	92.3	92.8	93.2
$[\sqrt[3]{n}]$	95.4	96.0	96.5	97.1	97.9	89.5	89.8	90.2	90.9	92.0
$[\sqrt{n}]$	89.0	89.8	90.9	92.0	94.0	83.1	83.6	84.2	85.0	86.2
$[\sqrt[4]{n^3}]$	71.3	71.7	72.4	73.5	75.1	60.2	60.6	61.3	61.9	63.4
$[\sqrt[5]{n^4}]$	64.2	65.4	66.5	67.7	70.0	53.1	54.3	55.7	57.3	58.4
[n/2]	57.2	58.2	59.1	60.4	62.0	48.3	47.9	47.5	46.9	46.0
[3n/4]	39.9	38.2	36.5	34.9	32.1	33.0	32.3	31.7	30.9	28.1
n	27.4	26.1	25.0	22.3	19.4	23.1	21.5	20.7	18.2	17.0

Таблица 2

		До	ля зада	ач,		Доля задач,				
	p	ешаемі	ых опти	имальн	О	решаемых оптимально				
	с помощью алгоритма					с помощью алгоритма				
	α2 (в %)					α_3 (B %)				
r/n	50	75	100	125	150	50	75	100	125	150
2	82.0	82.6	83.4	85.1	87.6	74.3	75.1	75.8	76.6	78.2
$[\sqrt[3]{n}]$	79.0	79.5	80.1	81.5	83.9	71.2	72.0	72.9	74.5	76.0
$[\sqrt{n}]$	73.2	73.8	74.5	75.4	76.2	65.0	65.4	66.0	66.8	68.2
$[\sqrt[4]{n^3}]$	53.4	54.4	55.7	56.2	57.0	46.2	45.8	45.3	44.8	44.1
$[\sqrt[5]{n^4}]$	44.8	44.1	39.6	39.0	38.2	39.0	38.1	37.3	36.4	35.4
[n/2]	41.3	41.0	40.7	38.5	37.4	36.5	36.1	35.7	34.9	33.4
[3n/4]	25.6	25.1	24.5	23.8	22.1	21.0	20.4	19.7	18.2	17.4
n	14.6	13.2	12.5	10.3	9.1	10.2	9.6	8.7	7.3	6.0

Проанализировав табл. 1 и 2, можно заключить следующее:

- 1) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \ge 50$, находимых посредством алгоритма α_0 , составляет не менее 96.1%, если r = 2, не менее 95.4%, если $r = [\sqrt[3]{n}]$, не менее 89.0%, если $r = [\sqrt[5]{n^4}]$, не менее 57.2%, если r = [n/2], причем с ростом n их доли увеличиваются;
- 2) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \ge 50$, находимых посредством алгоритма α_0 , с ростом n, по-видимому, будет уменьшаться, если $r \ge [3n/4]$;
- 3) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \ge 50$, находимых посредством алгоритма α_1 , составляет не менее 90.1%, если r = 2, не менее 89.5%, если $r = [\sqrt[3]{n}]$, не менее 83.1%, если $r = [\sqrt[4]{n^3}]$, не менее 53.1%, если $r = [\sqrt[5]{n^4}]$, причем с ростом n их доли увеличиваются;
- 4) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \geq 50$, находимых посредством алгоритма α_1 , с ростом n, по-видимому, будет уменьшаться, если $r \geq [n/2]$;
- 5) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \ge 50$, находимых посредством алгоритма α_2 , составляет не менее 82.0%, если r = 2, не менее 79.0%, если $r = [\sqrt[3]{n}]$, не менее 73.2%, если $r = [\sqrt{n}]$, не менее 53.4%, если $r = [\sqrt[4]{n^3}]$, причем с ростом n их доли увеличиваются;
- 6) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \geq 50$, находимых посредством алгоритма α_2 , с ростом n, по-видимому, будет уменьшаться, если $r \geq \lceil \sqrt[5]{n^4} \rceil$;
- 7) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \ge 50$, находимых посредством алгоритма α_3 , составляет не менее 74.3%, если r = 2, не менее 71.2%, если $r = [\sqrt[3]{n}]$, не менее 65.0%, если $r = [\sqrt{n}]$, причем с ростом n их доли увеличиваются;
- 8) доля оптимальных решений 4-аксиальной ПВ порядка n, $n \geq 50$, находимых посредством алгоритма α_3 , с ростом n, по-видимому, будет уменьшаться, если $r \geq [\sqrt[4]{n^3}]$.

Так как нами учитывались лишь те оптимальные решения 4-аксиальной ПВ, для которых значения целевой функции f(X) было равно n, то доли оптимальных решений, находимых посредством алгоритмов α_0 , α_1 , α_2 и α_3 , будут большими по сравнению с указанными в табл. 1 и 2. Поэтому по результатам вычислительных экспериментов можно сделать следующие выводы: алгоритм α_0 (соответственно α_1 , α_2 и α_3) почти всегда строит асимптотически оптимальный план 4-аксиальной ПВ, если $r \leq \lceil n/2 \rceil$ (соответственно $r \leq \lceil \sqrt[5]{n^4} \rceil$, $r \leq \lceil \sqrt[4]{n^3} \rceil$ и $r \leq \lceil \sqrt{n} \rceil$).

Список литературы

- 1. Balas E., Salthzman M. J. Facets of the three-index assignment polytope. // Discrete Appl. Math. 1989. Vol.23, N2. P. 201-229.
- 2. Arbib C., Pacciarelli D., Smriglio S. A. Three-dimensional matching model for perishable production scheduling // Discrete Appl. Math. 1999. Vol.92, P.1-15.
- 3. Robertson A. J. A set of greedy randomized adaptive local search procedure implementations for the multidimensional assignment problem // Comput. Optimiz. and Appl. 2001. Vol.19, P.145-164.
- Кравцов В. М. Полиномиальные алгоритмы нахождения асимптотически оптимального решения многоиндексной аксиальной проблемы выбора. // Кибернетика и системный анализ. 2005. №6. С.176-181.
- 5. Дичковская С. А. Вычислительное исследование приближенных и точных алгоритмов решения трехиндексной аксиальной проблемы выбора // Статья деп. в ВИНИТИ 09.06.04; N979-B2004. 33c.

Введение в автоматический синтез программ

Красилов А. А.

Отдел интеллектуальных систем 125252, Москва, Волоколамское шоссе, д.1, оф 412/1 E-mail: krasilov@galaktika.ru

Основной вопрос синтеза программ - это вопрос спецификация данных, по которым строится программа. Созданию спецификации программ предшествует работа Интеллсист – интеллектуальной системы общего назначения. Именно, решение запроса пользователя системой составляет спецификацию будущей программы. Рассматривается последовательность действий, приводящая к спецификации программы, а также языковые, логические и технические основы синтеза программ.

1. Путь к автоматическому синтезу программ

Для реализации синтеза программ необходимо было проделать ряд работ, который приведен ниже. Достаточно подробное описание работ можно увидеть в [1]. Результаты некоторых работ можно отнести к частично новому материалу в информатике. В конечном счете, нам необходимо сформулировать ту спецификацию задачи, которая может служить исходными данными для синтеза программ.

- Новое определение информатики, где вопросы обработки информации заменены вопросами обработки знаний;
 - Определение понятия знание [2];
 - Выделение семи видов представления знания [2] для приближения к естественному языку (ЕЯ);
- Представление и внедрение понятия термина вместо понятия идентификатора, что также обеспечивает приближение к ЕЯ:
- Построена семантическая грамматика русского языка (СеГ), содержащая правила вычисления смысла утверждений [3] (в отличие от известной всем классической синтаксической грамматики (СиГ), содержащей правила построения слов и предложений);
 - Понимание и обобщение понятия логического уравнения;
 - Построение информатической логики как основания для решения логических уравнений [4];
 - Построение методов решения логических уравнений (алгебраический подход в логике);
- Определение отладки и экологии знания (13 классов ошибок [5], 8 классов известных и 5 классов новых в программировании);
- Разработка формальных языков (ФЯ) Лейбниц для внутреннего представления знаний и метаязык Марков для представления порождающих грамматик;
- Разработка основ распознающих грамматик для автоматического построения программ, управляемых ФЯ (трансляторов и интерпретаторов);
 - Уточнение нового понятия интеллектуальной системы, именуемой Интеллсист;
- Определение понятия интеллектуального программирования (ИП) и критика процедурного программирования;
- Определение и реализация нового понимания синтеза программ, подпрограмм и пакетов подпрограмм;
- Конструирование программного комплекса инструментария ИП, доступного любому пользователю;
 - Разработка и развитие методов добычи знаний и построение БЗ [6].

Для разработки метода синтеза программ практически необходимо коснуться каждого пункта показанного пути.

Прежде всего, повышение интеллекта общения с ВМ достигается наилучшим приближением передаваемых ВМ записей знаний к ЕЯ. С формальной точки зрения для обобщения постановки задач вводится понятие логического уравнения (ЛУ), которое обобщает всю классификацию уравнений и неравенств, известных в математике. Для Интеллсист задаются тексты знаний в виде последовательностей равенств, неравенств или логических утверждений с любыми (из 150) операциями над величинами 36 типов, которые представлены на ЕЯ специалиста и образуют базы знаний (БЗ). Аналогично формируются запросы, требующие разрешения. БЗ и запрос образуют после перевода на ФЯ фундаментальное ЛУ, которое решается в Интеллсист. Следует учитывать при этом одно важное обстоятельство. БЗ - это класс решаемых заданий по области знаний, а запрос - конкретное задание на решение конкретной задачи.

Метод решения ЛУ определяется логикой, которая именуется информатической. Она обобщает все существующие логики и содержит более 500 аксиом (без явно заданной аксиомы дедукции) и правил

вывода решений. ИЛ определяет основания информатики и технологию (метод решения) обработки знаний [4]. Формально ЛУ записывается так:

где K_i - абзац БЗ, I - абзацы запроса, N - размер базы знаний. Каждая запись представляет обобщенное ЛУ. Аксиоматика информатического исчисления полностью определяет метод решения ЛУ путем логического и алгебраического вывода решения обобщенного ЛУ.

Исходный текст знаний и запросов должен соответствовать СеГ русского (или национального) изыка [3], содержащей правила вычисления смысла фраз и текстов. Она отличается от СиГ, состоящей из правил формирования слов и предложений. СеГ определяет правила и алгоритмы вычисления смысла фраз и текстов, которые обеспечивают понимание ВМ смысла знаний и запросов. Исходным в СеГ является понятие, представляемое парой имя и смысл, или термин и значение. Термин - это последовательность слов, знаков и лексем, только он имеет или может иметь реализацию конкретного смысла или значения. Последовательность терминов составляет фразы, из которых составляется текст, осмысливаемый Интеллсист. Совокупность текстов, составленных по правилам СеГ (и СиГ), является частью ЕЯ и именуется языком профессиональной прозы (ЯПП).

Основным вопросом в разработках Интеллсист является вопрос представления знаний. В разработках ИИ для представления знаний используются главным образом семантические сети и фреймы, а также специальные ФЯ. В реализации Интеллсист использовано универсальное представление знаний парами <имя, значение>, которая интерпретируется по различному в зависимости от изучаемого объекта. В СеГ она приобретает вид <термин, лексический элемент>, в СиГ - <текст, граф>, в графической информатике - <текст, изображение>. Можно представить и другие варианты интерпретации пары. Например, биологические исследования могут быть связаны с парой <человек, система>, лингвистические - с парой <язык, грамматика>, в автоматическом управлении — с парой <поведение, алгоритм>. Важно то, что пара отражает понятие некоторой области знания, имеющей две главные компоненты: наименование понятия и его смысл. Кроме этой пары каждое понятие имеет еще многие атрибуты, которые играют меньшую роль для пользователя и реже применяются. Примерами атрибутов понятия являются дата появления понятия, характеристика области смысла понятия и др.

Процесс формирования записи ЛУ называется ИП, которое не выдвигает требований к знанию приемов процедурного программирования. ИП можно (не всегда) отнести к ветви логического программирования. Это не всегда возможно, поскольку логическое программирование требует учета дедукции как метода логического вывода.

2. Представление знаний на ЯПП

Знания выражаются на ЯПП (да и на ЕЯ) в одной из форм:

- 1) Факт указанием имени и текущего значения (X = 1);
- 2) Утверждение относительно фактов или уже известных утверждений (X = 2 и Y = 3);
- 3) Правило получения нового факта или утверждения из известных фактов или утверждений (если X=0, то $\sin X=0$).

Указанных трех форм достаточно для представления любых знаний. При этом необходимо иметь в виду, что таблицы представляют фактический материал, тогда можно столкнуться с фактом СПИСОК ФАМИЛИЙ = [Иванов, Петров] или с утверждением СПИСОК ФАМИЛИЙ [1] = Иванов. Такие же рассуждения касаются массивов и записей.

Формы представления знаний относятся к использованию символов. Графические или предметные представления, можно сказать, не сводятся к символьным. Подобное сведение является отступлением от ЕЯ.

Из семи видов представления знаний ЯПП учитывает представление широко распространенного алгоритмического знания. Такие представления полезны, поскольку некоторые действия можно легко изобразить в форме алгоритма, нежели текстуально через утверждения. Например, построение таблицы простых чисел методом решета Эратосфена легче представить алгоритмом, нежели логической формулировкой. В качестве средств представления алгоритмов в ЯПП использована операция «.^» - следование действий без изменения порядка, и выражение «перейти к» для перехода по метке, которая может помещаться перед любым выражением.

Такие символьные представления используются для записей класса задач – спецификация класса или конкретной задачи спецификация задачи.

3. Экологическое обеспечение знаний

Процессы формирования текстов знаний и запросов составляют сущность ИП. С помощью ИП можно решать **7 классов задач**:

1. проверка истинности утверждений (или доказательство теорем),

- 2. поиск условий истинности (или синтез теорем и получение альтернативных решений, что отличает ИП от $\Pi\Pi$),
 - 3. определение или доопределение понятий,
 - 4. информационный поиск по смыслу сообщений или по таблицам,
 - 5. поиск пути логического вывода (или задачи лабиринта и планирования),
 - 6. синтез алгоритмов или программ,
 - информационный синтез систем.

Внешне нет различий в изображении задач. Отличие может быть при настройке инструментария. Эти классы охватывают все области деятельности пользователя. Почти каждое задание является композицией указанных задач, поэтому разбиение их на классы условно, чтобы обрамить область применения Интеллсист.

Структура Интеллсист и технология решения заданий представлена в [7]. В процессе анализа текстов и решения задачи выявляется **13** классов ошибок (около 250 конкретных названий). 8 важных и общих классов ошибок (синтаксические, лингвистические, концептуальные, семантические, прагматические, формульные, системные и предметные) широко известны, они обнаруживаются в каждой программной системе переработки текстов. Остальные 5 классов ошибок (смысловые, локальные и глобальные противоречия, независимость и полнота знаний) обнаруживаются в Интеллсист автоматически. Это отличает Интеллсист от экспертных и прологовых систем (ЭС и ПС). Автоматическое обнаружение 13 классов ошибок определяет экологическую чистку или контроль знаний и запросов.

В связи с этим родилась парадигма экологии знаний [5]. Важно определять чистоту знаний в текстах, что означает определить противоречия, независимость и полноту знаний. Независимость знаний обеспечивает фильтрацию данных и утверждений с целью экономии памяти и времени. Полнота знаний необходима для получения окончательного результата решения заданий после прохождения их через Интеллсист. Неполнота знаний также может оказаться полезной для формирования понятий, аналитических выкладок и синтеза программ. Локальные противоречия помогают проведению чистки знаний, а глобальное противоречие останавливает процесс решения задачи до выяснения причин таких противоречий. Такие проверки знаний отсутствуют в программировании другого сорта. Например, аналогичные проверки отсутствуют в ЭС.

4. Инструментарий Интеллсист

Разработка **инструментария ИП** сегодня именуется некоторой версией. Инструментарий инсталлирует (инициирует) работу Интеллсист, которая состоит из следующих частей:

- ядро (транслятор текстов с ЕЯ на внутренний формальный язык, интерпретатор операций для реализации смешанных вычислений, машина логического вывода решения логического уравнения, оформители ошибок и результатов и др.);
- интерфейс пользователя с Интеллсист (главная часть инструментария ИП) для обмена данными и знаниями;
 - Фактохран (соединение Интеллсист и СУБД), использование БД;
- **СБЗ** (встроенные и стандартные БЗ: Фундаментальные знания школа+ВУЗ, Физические единицы учет размерности величин, таблица Менделеева свойства элементов и др.);
 - Библиотека материалов (анкет, шаблонов, заданий и программ).
- ПБЗ (прикладные или заказные базы знаний по разделам наук, специальностей, отраслей знания или по специализации),
 - Фактонет (Интеллсист + Интернет для выборки в Фактохран) и справочная подсистема.

Кроме этого инструментарий имеет подсистемы настройки на проблему и на национальный язык, иллюстрации таблиц слов и терминов, справочную подсистему и подсистемы некоторых методов добычи значий

Важной особенность инструментария является следующее. ЯПП имеет разделы и некоторые формальные конструкции. Пользователь при наборе или вводе знаний или запроса набирает только термины, лексемы и нажимает на кнопки, а структуры формируются автоматически инструментарием.

5. Основы информатического исчисления

Информатическое исчисление (ИЛ) содержит более 500 логических аксиом и правила логического вывода (реализовано несколько тысяч аксиом по всем семи классам), которые вместе с ЯПП составляют систему для решения логического уравнения. Количество аксиом определяется числом встроенных операций и типов данных. Система может модифицироваться базой знаний и запросом, поскольку они изменяют ИЛ в части аксиом и правил вывода. Число аксиом не должно отпугивать читателя, если ему понятно существо аксиомы или исчисления: аксиома - это определение константы, величины, операции и связей между ними. Аксиомы, описывающие объекты и операции и представляющие знания, подразделяются на классы, которые являются взаимосвязанными, взаимозаменяемыми и взаимопереводимыми. Аксиомы всех классов учитываются при построении систем. Некоторые аксиомы

учитываются в процедурах обмена данными, другие - при построении алгоритмов или программ, третьи - явно входят в ИЛ, четвертые - присущи конструкции ВМ и т.п. Все аксиомы подразделяются на 7 классов. Рассмотрим общую характеристику каждого класса.

Класс естественных аксиом. Естественные аксиомы являются наиболее сложными, они определяют правильность, например восприятия знаков, их взаимосвязи и соответствия реальности. Ими пользуется человек или читающее устройство. Правильность использования характеризуется надежностью системы восприятия. Будем считать, что всегда имеется стремление к истине. Если она не достижима, то будем приближаться по возможности ближе к истине и наилучшим образом. Всегда главным считаем получение **истины**. К естественным аксиомам относятся, например, следующие:

- всегда рассматривается конечный набор символов (учитывается в кодировках);
- каждый символ должен отличаться от другого символа (символы различаются кодами);
- для символов действуют законы одинаковости;
- символ равен самому себе;
- символ имеет конечные размеры.

Естественные аксиомы всегда описательны, они могут задаваться алгоритмами. Их использует человек повседневно в силу накопленного в мозгу знания. Но описательные аксиомы можно формализовать или представить в каком-либо виде, например в заданном лексиконе. Данный класс аксиом может быть представлен в форме, которая свойственна любым другим классам. Они (как и все остальные классы) могут вводиться в БЗ для исследования объектов, которые определяются аксиоматически, они могут в каком-либо виде входить в запросы.

Класс языковых аксиом. Языковые аксиомы определяют символьные комбинации с помощью формальных грамматик, которые сами являются собранием аксиом, и реализуются в системах обработки входных сообщений трансляторами и интерпретаторами, они именно там и учитываются. Часть уже формализованных аксиом можно найти в [8] относительно структурных языковых объектов. Например, в указанной замечательной монографии определяются аксиомы для перечислимых, записей, массивов, Рмножеств и последовательностей. Там представлена полная система аксиом для операции катенации, которая отсутствует в системе логических аксиом из-за свойств памяти ВМ (разумеется, и человека) сохранять и последовательно обрабатывать символы или их коды. Еще пример. Запись данных контролируется на правильность подбора значений для полей. Средства контроля следуют из аксиом для значений типа записи. Такие аксиомы определяют смысловую сторону объектов и операций. Они в обязательном порядке используются при построении транслирующих, компилирующих или интерпретирующих систем.

Класс концентуальных аксиом. Концептуальные аксиомы определяют понятия, необходимые для передачи знаний в память ВМ. В качестве примеров таких аксиом можно указать на аксиомы выбора атрибута, индексирования, именования, квалификации, преобразования типов, аксиомы для процедур и присваивания, обычно вводимых в описаниях семантики языков программирования. Лексикон является собранием концептуальных аксиом. Примеры конкретных аксиом приведены ниже:

- число индексов у переменной должно равняться числу граничных пар в описании массива;
- указанию имени поля записи должно следовать за именем записи;
- типы левой и правой частей присваивания должны быть одинаковыми.

Большинство концептуальных аксиом работают при вводе информации в память ВМ или Интеллсист или при изучении контекстов для объектов (терминов).

Класс аксиом для наборов (или аксиомы структур) состоит из аксиом, которые связаны с данными и массивами данных. Эти аксиомы определяют наборы данных или элементарных знаний с точки зрения структуры и свойств всей совокупности данных и знаний. Они отражают прагматику данных или каких-либо языковых конструкций. Такие аксиомы учитываются в семантических преобразованиях или в алгоритмах обработки данных.

Класс логических аксиом. Логические аксиомы определяют математические константы, величины (свойства предметной области) и операции (свойства проблемной области) и их свойства для обработки знаний и запросов с целью логического вывода нового знания. Они должны быть формальными и основываться на исследованиях математики, а точнее - на исследованиях математической логики. Они являются исходным знанием для заполнения стандартной части БЗ, которая реализуется в виде МЛВ и МАВ – машин логического и алгебраического вывода (встроенные знания).

Класс прикладных аксиом. Прикладные аксиомы определяют объекты и операции пользователя, они являются исходным знанием в данной области или специализации для заполнения пользовательской части БЗ и «программой» решения класса задач. Пользователь передает свои знания или запросы в большинстве своем прикладными аксиомами (утверждениями об исследуемых объектах или операциях). Собственные знания пользователя содержат многочисленные аксиомы, определяющие его объекты и операции. Любая запись знаний или запросов являются набором прикладных аксиом.

Класс системных аксиом. Системные аксиомы определяют операции ВМ и ОС по вводу исходных данных, обработке и выводу итоговых представлений знаний (данных). Их использование осуществляется при разработке Интеллсист и ее инструментария. Большая часть системных аксиом

реализована в операционных системах и системных программах. Можно сказать, что системные аксиомы автоматически учитываются при работе с ВМ или Интеллсист.

Итак, в ИЛ каждая аксиома - это определение объекта или операции, термина. Любое определение устанавливает взаимосвязь объектов и операций, а также включает неопределенные аксиомой данного исчисления (или ИЛ) объекты и операции. Предполагается, что неопределенные объекты и операции описаны с помощью аксиом другого класса. Они составляют объекты для поиска. Указанные положения можно использовать при конструировании аксиом исчисления. Вообще же, исчисление ИЛ опирается на язык Лейбниц, в котором зафиксирован набор типов данных и операций.

Некоторые аксиомы служат проверке правильности выражений или вычислений, некоторые аксиомы предназначены для обнаружения и локализации ошибок. Большая часть аксиом является основой для формирования алгоритмов или процедур логического или алгебраического вывода, вычислений или для формирования печати результатов.

6. Исходы работы Интеллсист

Уже понятно, что исходная спецификация программы составляется из результатов прогона запроса и БЗ через ядро системы. Действительно, Обобщенное ЛУ обрабатывается ядром до получения альтернатив совместных фактов или утверждений. В такую композицию входят последовательность фактов или утверждений, которые образуют список через запятую (совместность) и перечни через точку с запятой альтернатив из списков. Такая композиция является спецификацией будущей программы. Например, если кроме фактов в решении имеются утверждения, то это означает, что не хватает данных для получения окончательного решения.

Еще раз. Факт - это либо элементарное высказывание типа «солнце встало» или «X=7.0», либо утверждение (определение понятий или итоги аналитических выкладок). Если исходные данные не заданы, то по утверждениям синтезируется программа. Недостающие данные могут вводиться в программу, а после ее выполнения могут получиться окончательные факты. Если опять не хватает данных, то результатом работы программы будут аналитические выражения, которые могут составлять материал для новых методов решения или алгоритмов.

Инструментарий ИП является средством решения заданий, написанных на ЯПП, которое доступно любому пользователю, знающему лексикон и проблематику своей профессии. Для пользователя продуктами всей разработки являются настроенная Интеллсист, встроенная в систему БЗ, фундаментальная БЗ, БЗ физических единиц, интерфейсная БЗ для синтезированных программ и заказные БЗ. Использование Интеллсист не требует знаний формальных языков и услуг программистов. Ныне компьютерный рынок поставляет (кроме технических средств) программы, которые представляют алгоритмическое знание, а БЗ рождают новую нишу на рынке машинных знаний.

Итак, результатами работы Интеллсист являются:

альтернативы новых совместных фактов, программа для вычислений, ошибки в задании. Ошибки порождают циклы исследований или отладку знаний.

7. Развитие и перспективы Интеллсист

Использование БД в Интеллсист предусмотрено в двух вариантах. Первый - это создание различных таблиц и коллекций данных, собираемых по знаниям и запросам. Данные являются фактическим материалом (фактами) для логического вывода ответов на запросы. Второй - это использование стандартных СУБД или наборов различных СУБД (по выбору пользователя), интеллектуальность которых заложена заведомо разработчиками выбранных СУБД.

 $И\Pi$ в 21 веке станет основным средством решения задач непосредственно пользователем вычислительной техники наравне со всепроницающим Интернет'ом, обеспечивающим поиск готовых решений. Лозунг программистов о том, что все можно запрограммировать остается актуальным и сегодня с некоторой модификацией. Компьютер может решать любые задачи на основе знаний, представленных на $E\Pi$, без привлечения программистских сил и $\Phi\Pi$.

Для внутреннего представления знаний и запросов разработан формальный язык Лейбниц, грамматика которого универсальна и не зависит от конкретного ЕЯ. Он формализует внутренние представления текстов знаний и запросов без операторных структур и не является обязательным для пользователя.

Пользователь общается с инструментарием ИП через несколько окон из числа имеющихся 150-200 окон, получая ответы на запросы о параметрах и режимах работы Интеллсист (применение параметров по умолчанию сокращают работу), о файлах для работы, редакторских операциях, способах индикации окон, необходимом справочном материале для обучения, сервисе обработки знаний, структуре знаний и свойствах вводимых понятий и др. Основная работа состоит в создании лексикона, БЗ, запроса и его разрешении. Справочная подсистема включает стандартный терминологический словарь по информатике обработки знаний, индекс, общую информацию об Интеллсист. Среди сервисных важной является программа сборки терминологического словаря по исходным текстам знаний (учебникам, пособиям и руководствам). Первичная обработка текстов в результате корреляционного анализа их по

смыслу определяет черновой терминологический словарь [6] в помощь пользователю для создания фактических словарей и при редактировании исходных текстов знаний в соответствии с правилами СеГ и выделения терминов. Кроме указанного в состав сервиса входит программа контроля грамматического правописания (терминография).

Интеллектуальность Интеллсист оценивается, как и другие программы по семи номинациям:

- (1) наглядность или внешняя интеллектуальность, не требующие емких руководств, интерфейс обеспечивает подсказки в работе (по этой номинации ныне все программы оцениваются высоко),
- (2) осмысленность, внутренняя интеллектуальность, оснащенность языками для решения лингвистических подзадач (трансляторы, распознаватели, использование СеГ),
- (3) концептуальность или базовые основы программного комплекса (наличие словарей и лексиконов для понятий),
- (4) использование БД для взаимосвязи данных, их целостности и защищенности,
- (5) наличие логического вывода или вывод ответов на запросы пользователей по БЗ,
- (6) программная пригодность, возможность настройки и адаптации программы,
- (7) системная пригодность или общественная жизнь программного комплекса.

Во многих программах большая часть номинаций учитывается в той или иной мере, кроме пункта 5 (исключая ЭС и ПС). Интеллсист проходит по всем номинациям позитивно и с высоким рейтингом.

Оценка системы обработки знаний по четвертой номинации целиком зависит от используемых СУБД и версии Интеллсист, отличающейся от других мерой использования каждой СУБД или их набором. Шестая номинация связана с удобствами в программировании: язык и стиль, возможность создания больших программ, надежность и эффективность программ и процессов программирования, определяемая синтезатором программ, возможность тестирования, верификации и подтверждения знаний. Интеллсист предусматривает автоматический синтез программ, что обеспечивает их хорошие характеристики, построение программ средствами ЯПП. Седьмая номинация предусматривает и ориентирует разработчиков знаний и программ по многим атрибутам системного порядка: рынок, маркетинг, реклама, аудит, конкурентоспособность. Относительно Интеллсист пока рано говорить о высокой оценке интеллектуальности по этой номинации. Критерием здесь явится размер спроса. Многие программные комплексы заслужили наивысшие оценки по перечисленным номинациям. Их разработка и распространение достойны внимательного анализа и наиболее широкого применения. Совокупная оценка по всем номинациям каждого программного комплекса является средством для сравнения интеллектуальности программ между собой. Она может быть получена как среднеарифметическая или усредненная по оценкам всех номинаций.

Общее руководство пользователя по инструментарию содержит около 450 стр. Отдельно рассматривается описание основ информатики в 7 томах (2100 страниц) для обоснования нового определения науки информатика. На рынок будет представлен комплекс инструментария вместе с Интеллсист. Кроме этого на рынке начнут появляться БЗ по отраслям знаний для решения прикладных задач (новая нища ранка программ).

Новые интеллектуальные системы получаются следующим образом. Инструментарий Интеллсист среди прочих операций содержит операцию композиции готовых Интеллсист, аналогичные операциям формирования новых алгоритмов из известных. Необходимыми действиями для реализации этого являются операции над лексиконами и БЗ.

Список литературы

- 1. www.intellsyst.ru
- 2. Красилов А. А. Основы информатики. Определение и концепции. // Учебное пособие, МФТИ, Долгопрудный, 1990, 80 с.
- 3. Красилов А. А. Язык и знание. Семантическая грамматика русского языка. //Научно-техническая информация, серия 2, № 10, ВИНИТИ, 2000, с 21-33.
- 4. Красилов А. А. Информатическая логика, решение логических уравнений и интеллектуальные системы в САПР. Информатика. //В сб: Автоматизация проектирования, вып.2-3, ВНИИМИ, М., 1994, с.14-26
- 5. Красилов А. А Экология информации и знаний. Научно-техническая информация, серия 1, Организация и методика информационной работы. № 2. М., ВИНИТИ, 2005, с. 11-23
- 6. Красилов А. А. Knowledge Mining от сырых знаний к полезным решениям и новым знаниям. «Аналитика-капитал», том XI «Генезис информатики и аналитики в корпоративном и административном управлении», М., ВИНИТИ, 2005, с. 205-225
- 7. Красилов А. А., Григорьев Р. Д. Технология интеллектуального программирования. //Сб. докл.: Высокие технологии в промышленности России, -М., Международная академия информатизации, 1999, с.; и //Инженерный журнал, Справочник, N 10 (31), Машиностроение, 1999, с. 34-37.
- 8. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М., Мир, 1975, 248 с.

Автоматы в лабиринтах

В. Б. Кудрявцев, Г. Килибарда, Ш. Ушчумлич

В последние десятилетия все большее внимание привлекает тематика, связанная с автоматным анализом геометрических сред, изображений, графов, формальных языков и других дискретных структур. Интерес к этому кругу проблем особенно усилился с развитием кибернетики. Сформировалось новое направление в теории автоматов — поведение автоматов в лабиринтах. В работе анализируется развитие этого направления и дается обзор его основных результатов.

Важную роль в становлении этого направления сыграла работа К. Шеннона [37]. В ней рассматривается модель мыши в виде автомата, которая должна найти определенную цель в лабиринте. Это в значительной мере определило тематику исследований на многие годы.

Другим источником направления можно считать рассмотрение вычислительных систем с внешней памятью в виде дискретной плоскости или дискретного пространства П. Фишера [14].

Однако идеи, предложенные в этих работах, довольно долго не получали дальнейшего развития. Это было связано с рядом причин. В первом случае автомат-мышь действовал в геометрической среде, в то время как еще не вполне было ясно, как автомат преобразует свою входную полугруппу слов, не имеющих никакой интерпретации. Выяснению этого и были посвящены основные усилия исследователей в области теории автоматов.

Во втором случае внимание специалистов быстро было переключено на многоленточные вычислители, что и вытеснило модель Фишера.

Активное изучение поведения автоматов в лабиринтах и на графах началось после появления работ К. Деппа [11,12]. В них была уточнена модель К. Шеннона: в качестве лабиринта рассматривалась подобная шахматной доске конфигурация клеток на плоскости или аналогичных кубиков в пространстве (шахматные лабиринты), а в качестве автоматов — конечные автоматы, которые, обозревая некоторую окрестность клетки, в которой они находятся, могут перемещаться в соседнюю клетку в одном из координатных направлений. В [11,12] получены некоторые результаты по задаче обхода таких лабиринтов автоматами, и как актуальный выделен вопрос о существовании автомата, обходящего все лабиринты; приведены соображения в пользу того, что в случае пространственного лабиринта для автомата можно построить лабиринт-ловушку. Х. Мюллеру [29] удалось построить для заданного автомата плоскую ловушку в виде 3-графа, а Л. Будах [7] решил более сложный случай этой проблемы — построение шахматной ловушки для любого заданного конечного автомата, однако его обоснование оказалось слишком громоздким. А. С. Подколзин [51,52] значительно упростил доказательство этого факта, а Г. Килибарда [24,49] дает другое, методически более наглядное решение этой проблемы, техника которого позволила решить и некоторые другие задачи типа задач обхода. Х. Антельман [2] оценил сложность такой ловушки по числу клеток в ней, а Х. Мюллер [30] указал, что всегда в качестве ловушки можно выбрать r-связный лабиринт, причем $r \leq 3$. Затем Х. Антельман, Л. Будах и Х. Роллик [1] построили конечную ловушку для конечной системы автоматов и бесконечную ловушку сразу для всех конечных систем автоматов.

Характеризацию специальных типов графов-лабиринтов, "правильно" укладывающихся в плоскость, которые появляются в упомянутых выше работах, дали Ф. Хофман и К. Кригел [20], а также независимо от них подобную характеризацию получили Г. Вияян и А. Вигдерсон в работе [39].

Наряду с результатами, указывающими на ограниченность возможностей автоматов, были построены примеры классов лабиринтов, которые обходятся одним автоматом (Г. Ассер [3], Р. Данецки и М. Карпински [10], К. Депп [11,12], Г. Килибарда [46,48], Г. Килибарда и Ш. Ушчумлич [50] и др.). А. Н. Зыричев [45] установил, что класс всех плоских шахматных лабиринтов, имеющих дыры ограниченного диаметра, обходится одним автоматом. А. А. Золотых [44] расширил этот класс, показав, что можно рассматривать ограниченность дыр лишь по фиксированному направлению. В работах [44,45,48,50] содержатся также оценки времени обхода этих лабиринтов и числа состояний для автоматов.

Невозможность обхода всех плоских шахматных лабиринтов одним автоматом выдвинула вопрос об изучении возможных усилений модели автомата, уже решающей задачу обхода. Исследовано несколько вариантов таких усилений. Главным из них является система взаимодействующих автоматов, называемая коллективом. В отличие от системы независимых автоматов коллектив анализирует лабиринты с учетом положения его "членов" в них. Другое усиление возможностей одного автомата восходит к Р. Фишеру [14] и состоит в разрешении автомату делать отметки в лабиринте. В случае, когда такие отметки могут ставиться и стираться, получаем машину Тьюринга общего типа и потому рассматриваемые здесь задачи легко решаемы. Значит, надо ограничивать право автомата на отметки. Здесь возникли случаи нестираемых, иерархичных, а также "мерцающих" отметок и др., [41,42,55,57]. Содержательно, возможность делать такие отметки означает, что автомат обладает внешней неограниченной памятью. Таким образом, как и следовало ожидать, возможности

автоматов резко усилились.

Оказалось, например, что для заданной размерности пространства n и k отметок существует автомат с k нестираемыми отметками, который обходит любой n-мерный прямоугольный лабиринт [56].

Кроме задачи обхода рассматривалась и другая, тесно связанная с ней проблема, — проблема остановки автомата в лабиринте. Она состоит в следующем: существует ли для некоторых содержательно интересных классов лабиринтов, имеющих хотя бы один универсальный обходчик, автомат, который после обхода любого лабиринта из этого класса останавливается в нем. М. Бул и А. Хемерлинг [8] показали, что при решении проблемы остановки, автоматы являются еще более бессильными моделями, чем при решении проблемы обхода.

Как выяснилось, независимая система автоматов оказалась слишком слабой при решении значительной части задач, возникающих в этой области. Поэтому предлагались различные усиления этой модели вплоть до модели машины Тьюринга (в ее обобщенном виде). Поскольку машина Тьюринга легко решает рассматриваемые данным направлением задачи, то самым интересным и главным усилением модели независимой системы автоматов оказывается коллектив автоматов.

В отличие от независимых систем автоматов коллектив автоматов анализирует лабиринты с учетом положения его "членов" в лабиринте. Коллективы автоматов могут быть снабжены камнями, играющими роль ограниченной внешней памяти для коллективов. Камень — это флажок-метка, который может быть или "на руках" у любого автомата, или в некоторой вершине обследуемого коллективом лабиринта. Автомат может забирать или оставлять камень в вершине, в которой оказался, но общее количество камней у данного коллектива всегда является постоянным. Камни можно рассматривать также как автоматы простейшего вида, перемещение которых полностью определяется другими автоматами коллектива. Ф. Хофман [18,19] установил, что коллектив из одного автомата и одного камня не может обойти все конечные плоские мозаичные лабиринты; М. Блюм и Д. Козен [6] показали, что коллектив из одного автомата и двух камней решает эту задачу, отметив при этом, что коллектив из двух автоматов также решает эту задачу. Наряду с этим в работах А. Хемерлинга [16] и К. Кригела [26] показано, что класс указанных лабиринтов допускает естественное расслоение, такое, что для любого его слоя найдется коллектив из одного автомата с камнем, обходящий этот слой; в качестве параметра расслоения здесь выступает число дыр в лабиринте.

Аналогичный вопрос для класса всех конечных и бесконечных плоских мозаичных лабиринтов исследуется в работах М. Блюма и У. Сакоды [5], З. Хабасинского и М. Карпинского [15], А. Шепитовского [38], Г. Килибарды [24,49] и др. В них установлены некоторые простейшие по числу автоматов и камней коллективы, обходящие все такие лабиринты. Г. Килибарда в работах [24,49] практически завершил описание всех таких коллективов (открытой проблемой остался только случай коллектива, состоящего из одного автомата и 4 камней); в этих же работах приведено решение указанной задачи для лабиринтов, не содержащих бесконечные дыры.

Для лабиринтов более общего вида М. Блюмом, У. Сакодой [5] и А. Хемерлингом [17] показано наличие ловушки уже в трехмерном случае. Установлено наличие бесконечной трехмерной ловушки сразу для всех коллективов автоматов [47]. При этом коллективы остаются в шаре ограниченного радиуса в этой ловушке. Подобные результаты оказываются верными и в планарном случае, для лабиринтов, имеющих вид кубического графа [34].

Специальными классами лабиринтов являются так называемые сигнатурные лабиринты и лабиринты Савича. Для первого вида в работе Г. Л. Курдюмова [?] описан коллектив автоматов, находящий специальную вершину в этих лабиринтах. Для второго вида установлено, что проблема выхода из них по специальным путям эквивалентна открытой проблеме совпадения языков, распознаваемых детерминированными и недетерминированными линейно ограниченными машинами Тьюринга, что свидетельствует о больших потенциальных трудностях тематики [35,36].

Начато исследование задачи о встрече коллективов автоматов в лабиринтах. Одним из возможных толкований этой задачи может быть описание для заданного класса лабиринтов всех пар коллективов, которые встречаются в любом лабиринте из этого класса. А. Пултр и Й. Улехла [33] показали, что два коллектива, каждый из которых состоит из одного автомата и двух камней, могут решить задачу о встрече на плоскости, а в работе А. В. Анджанса [40] указаны простейшие типы коллективов автоматов, решающих задачу о встрече на прямой и в плоскости.

В работах В. И. Грунской ([43]) рассмотрен вариант задачи о встрече двух автоматов, находящихся в отношении "хищник — жертва", где "автомат-хищник" пытается догнать "автомат-жертву", а "автомат-жертва" стремится убежать от него; взаимодействие происходит в квадратном лабиринте. Приводятся условия, при которых указанная встреча происходит. В работах У. Койа [9], М. Була и А. Хемерлинга [8], А. В. Анджанса [40] и других авторов рассматривались возможности более общих моделей автоматов в лабиринтах. Так, например, в работе [9] показано, что автомат с магазинной памятью не может обойти все лабиринты, имеющие вид кубических графов, а в [40]

приведены примеры автоматов со счетчиками, со стеками и магазинами, обходящих мозаичную плоскость. Установлено также, что существует автомат с магазином, который обходит любой конечный плоский односвязный шахматный лабиринт и останавливается после его обхода [8].

Задача анализа для автоматов и лабиринтов изучалась в работах многих авторов: М. Блюм и К. Хюит [4], М. Ейсмонт [13], К. Иноуе, И. Таканами и А. Накамура [21], К. Иноуе и А. Накамура [22], К. Иноуе и И. Таканами [23], Е. В. Кинбер [25], Д. Л. Милиграм и А. Розенфелд [27], Д. Л. Милиграм [28], Й. Милопулос [31], У. Савич [35,36] и др. Она состоит для заданного коллектива автоматов в описании всех лабиринтов, которые обходятся этим коллективом при возможных дополнительных соглашениях типа требования остановки после обхода. Попытки описать эти лабиринты в виде алгебры Клини встретили затруднения [21]; трудности возникли также при выяснении отношений между классами лабиринтов, представляющих решение задачи анализа для заданных коллективов автоматов [23]. В работе [4] показано, что классы лабиринтов, обходимые автоматами с камнями, неограниченно возрастают с увеличением числа камней. Анализу свойств нагруженных графов посвящена работа Г.Ю.Кудрявцева [53], в которой устанавливается, с какой сложностью может быть решена задача эквивалентности поведения автоматов в таких графах. В работах [27,32] устанавливается связь между классами лабиринтов и формальными языками, что приводит к переплетению проблематики и переходу к решению задач для автоматов в лабиринтах и для языков.

Как видно из перечисленных результатов основная проблематика для автоматов в лабиринтах группируется вокруг задач, условно называемых задачами синтеза и анализа.

Задача синтеза состоит в описании тех автоматов и коллективов автоматов, которые обходят лабиринты из заданного класса. Главными объектами здесь выступают конечные автоматы с камнями, коллективы автоматов, автоматы с магазинной памятью и другие, а также конечные или бесконечные плоские мозаичные лабиринты, различные классы таких лабиринтов (как, например, лабиринты ограниченной связности, специальной геометрической формы), плоские, но немозаичные, мозаичные в пространстве, конечные и бесконечные лабиринты и их подклассы и обобщения. В случае, когда для класса лабиринтов отсутствуют автоматы заданного типа, обходящие эти лабиринты, возникают такие задачи, как выделение тех лабиринтов, в которых автоматы не решают задачу обхода, т.е. лабиринтов-ловушек или просто ловушек, и нахождение среди этих ловушек таких, которые обладают некоторыми свойствами, например, являются простейшими.

Задача анализа, как уже было сказано выше, состоит в описании по заданному автомату или коллективу автоматов всех лабиринтов или всех лабиринтов определенного вида, которые обходятся этими автоматами. В качестве автоматов и лабиринтов выступают объекты, указанные в задаче синтеза. Продвижение в решении этой задачи в сравнении с задачей синтеза значительно скромнее и состоит, по-существу, в построении различных позитивных и негативных примеров возможных соответствий между конкретными множествами лабиринтов и автоматов, обходящих в точности эти лабиринты, или же отсутствие таковых и т.п.

К числу задач, примыкающих к указанным задачам синтеза и анализа, относится поиск конкретных целей в лабиринтах, например, специальных областей или других автоматов; установление определенных свойств лабиринтов, например, наличие специальных циклов, сильной связности и прочее. Сюда же могут быть отнесены и проблемы распознавания свойств геометрических изображений, допускающих описание с помощью формальных языков, и др.

В качестве главной модели выступают конечные автоматы в конечных плоских мозаичных и бесконечных плоских мозаичных лабиринтах.

Список литературы

- 1. Antelmann H., Budach L., Rollik H. A. On universale traps // EIK. 1979. Vol. 15, no. 3. Pp. 123–131.
- 2. Antelmann H. An application of the prime number theorem in automata theory // ICS PAS Reports 411.-1980.-Pp. 9–11.
 - 3. Asser G. Bemerkungen zum Labyrinth-Problem // EIK. -1977. Vol. 13, no. 4/5. Pp. 203–216.
- 4. Blum M., Hewitt C. Automata on a two-dimensional tape // Proc. 8th IEEE SWAT Conf. 1967. Pp. 155–160.
- 5. Blum M., Sakoda W. On the capability of finite automata in 2 and 3 dimensional space // Proc. 17th IEEE FOCS Conf. 1977. Pp. 147–161.
- 6. Blum M., Kozen D. On the power of the compass // Proc. 19th IEEE FOCS Conf. 1978. Pp. 132–142.
 - 7. Budach L. Automata and labyrinths // Math. Nachrichten 86. 1978. Pp. 195–282.

- 8. Bull M., Hemmerling A. Finite embedded trees and simply connected mazes cannot be searched by halting finite automata // Journal of Inf. Process. and Cybern. EIK. 1990. Vol. 26, no. 1/2. Pp. 65–73.
 - 9. Coy W. Of Mice and Maze // EIK. 1978. Vol. 14, no. 5 Pp. 227–232.
- 10. Danecki R., Karpinski M. Decidability results on plane automata searching mazes // Proc. 2nd Int. FCT'79 Berlin: Conf. Akademie Verlag, 1979. Pp. 84–91.
 - 11. Döpp K. Automaten in Labyrinthen I // EIK. 1971. Vol. 7, no. 2. Pp. 79–94.
 - 12. Döpp K. Automaten in Labyrinthen II // EIK. 1971. Vol. 7, no. 3. Pp. 167–190.
- 13. Ejsmont M. Problems in labyrinths decidable by pebble automata // EIK. 1984. Vol. 20, no. 12. Pp. 623–632.
- 14. Fischer P. C. Multi-tape and infinite-state automata: A survey // Comm. ACM. 1965. Vol. 8, no. 12. Pp. 799–805.
- 15. Habasinski Z., Karpinski M. A codification of Blum-Sakoda 7-pebbles algorithm // ICS PAS Reports 448. Warszawa, 1981.
- 16. Hemmerling A. 1-pointer automata searching finite plane graphs // Z. Math. Logik Grundlag. Math. 1986. Vol. 32. Pp. 245–256.
- 17. Hemmerling A. Normed two-plane traps for finite systems of cooperating compass automata // J. Inf. Process. Cybern. EIK 1987. Vol. 28, no. 8/9. Pp. 453–470.
- 18. Hoffmann F. One pebble does not suffice to search plane labyrinths // Lecture Notes in Computer Science. 1981. Vol. 117. Pp. 433-444.
- 19. Hoffman F. 1-Kiesel-Automaten in Labyrinthen // Report R-Math-06/82. AdW der DDR, Berlin, 1982.
- 20. Hoffman F., Kriegel K. Quasiplane labyrinths. Preprint / P-Math-20/83. AdW der DDR, Berlin, 1983.
- 21. Inoue K., Takanami I, Nakamura A. A note on two-dimensional finite automata // Information Processing Letters. 1978. Vol. 7, no. 1. Pp. 49–52.
- 22. Inoue K., Nakamura A. Two-dimensional finite automata and unacceptable functions // International Journal of Computer Mathematics. Section A. 1979. Vol. 7 Pp. 207-213.
- 23. Inoue K., Takanami I. A note on decision problems for three-way two-dimensional finite automata // Information Processing Letters. 1980. Vol. 10. Pp. 245–248.
- 24. Kilibarda G. On the minimum universal collectives of automata for plane labyrinths // Discrete Math. Appl. 1993. Vol. 3, no. 6. Pp. 555–586.
- 25. Kinber E. B. Three-way automata on rectangular tapes over a one-letter alphabet // Information Sciences. -1985. Vol. 35. Pp. 61–77.
- 26. Kriegel K. Universelle 1-Kiesel-Automaten für k-komponentige Labyrinthe // Report R-Math-04/84. AdW der DDR, Berlin, 1984.
- 27. Miligram D. L., Rosenfeld A. Array automata and array grammars // IFIP '71 Conference Proceedings. North-Holland, Amsterdam, 1972. Pp. 69–74.
- 28. Miligram D. A region-crossing problem for array-bounded automata // Information and Control. 1976. Vol. 31, no. 2. Pp. 147–152.
 - 29. Müler H. Endliche Automaten und Labyrinthen // EIK. 1971. Vol. 7, no. 4. Pp. 261–264.
- 30. Müler H. Automata catching labyrinths with at most three components // EIK. 1979. Vol. 15, no. 1/2. Pp. 3–9.
- 31. Mylopoulos J. On the recognation of topological invariants by 4-way finite automata // Computer Graphics and Image Processing. 1972. Vol. 1. Pp. 308–316.
- 32. Mylopoulus J. On the application of formal languages and automata theory to pattern recognition // Pattern Recognition 1972. Vol. 4, no. 1. Pp. 37–51.
- 33. Pultr A., Úlehla J. On two problems of mice // Rend. Circ. Mat. di Palermo. 1982. Vol. 31, no. 2. Pp. 249-262.

- 34. Rollik H. A. Automaten in planaren Graphen // Acta Informatica. 1980. Vol. 13. Pp. 287—298.
- 35. Savitch W. Relations between nondeterministic and deterministic tape complexities // Journal of Computer and System Science. 1970. Vol. 4. Pp. 177–192.
- 36. Savitch W. Maze recognizing automata and nondeterministic tape complexity // Journal of Computer and System Science. -1973.- Vol. 7.- Pp. 389-403.
- 37. Shannon Cl. E. Presentation of a maze-solving machine // Cybernetics Trans. of the 8th Conf. of the Josiah Macy Jr. Found / Editor: H. Foerster. 1951. Pp. 173–180.
- 38. Szepietowski A. A finite 5-pebble-automaton can search every maze // Information Processing Letters. 1982. Vol. 15, no. 5. Pp. 199–204.
- 39. Vijayan G., Wigderson A. Rectilinear graphs and their embeddings // SIAM J. Comput. 1985. Vol. 14. Pp. 355–372.
- 40. Анджанс А. В. Поведение детерминированных и вероятностных автоматов в лабиринтах: Дисс... канд. физ.-мат. наук. Рига, 1987. 90 с.
- 41. Голованов А. В. Об обходе лабиринтов автоматами, оставляющими след в вершинах лабиринта // Интеллектуальные системы. 1998. T. 3, вып. 3-4. C. 193-212.
- 42. Голубев Д.В. Об обходе графов автоматами с одной нестираемой краской // Интеллектуальные системы. 1999. Т. 4, Вып. 1-2. С. 243—272.
- 43. Грунская В. И. О динамическом взаимодействии автоматов // Математическая кибернетика и ее приложения к биологии. М.: Изд-во МГУ, 1987. С. 8–18.
- 44. Золотых А. А. Обход лабиринтов с ограниченными в фиксированных направлениях дырами // Дискретная математика. 1993. Т. 5, вып. 1. С. 59—69.
- 45. Зыричев А. Н. О синтезе автомата, обходящего плоские лабиринты с ограниченными дырами // Дискретная математика. 1991. Т. 3, вып. 1. С. 105–113.
- 46. Килибарда Γ . Об универсальных лабиринтах-ловушках для конечных множеств автоматов // Дискретная математика. 1990. Γ . 2., вып. 1. Γ . 72–79.
- 47. Килибарда Γ ., Ушчумлич Ш. О лабиринтах-ловушках для коллективов автоматов // Дискретная математика. 1993. Т. 5, вып. 2. С. 29–50.
- 48. Килибарда Г. О сложности автоматного обхода лабиринтов // Дискретная математика. 1993. Т. 5, вып. 3. С. 116–124.
- 49. Килибарда Г. О минимальных универсальных коллективах автоматов для плоских лабиринтов // Дискретная математика. 1994. Т. 6, вып. 4. С. 133—153.
- 50. Килибарда Г., Ушчумлич Ш. О задаче синтеза для автоматов в одном классе лабиринтов // FILOMAT (Niš) 9:3 (1995), 743-751.
- 51. Кудрявцев В. Б., Подколзин А. С., Ушчумлич Ш. Введение в теорию абстрактных автоматов. М.: Изд-во МГУ, 1985.
- 52. Кdr АРКудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. — М.: Наука, 1985.
- 53. Кудрявцев Г. Ю. О времени решения лабиринтных задач конечными автоматами: Дисс... канд. физ.-мат. наук. Саратов, 1990.-127 с.
- 54. Курдюмов Г. Л. Коллектив автоматов с универсальной проходимостью // Проблемы передачи информации. 1981. Т. 17, вып. 4. С. 98–112.
- 55. Насыров А. З. Об обходе лабиринтов автоматами, оставляющими нестираемые метки // Дискретная математика. 1997. Т. 9, вып. 1. С. 123—133.
- 56. Насыров А. З. Об обходе автоматами лабиринтов в n-мерном пространстве // Дискретная математика. 2000. Т. 12, вып. 4. С. 121–137.
- 57. Насыров А. З. О бесконечной ловушке для автоматов со следом // Интеллектуальные системы. -2000. Т. 5, вып. 1–4. С. 273–285.

Применение технологии углубленного анализа данных (data mining) для построения системы информационного мониторинга риска атеросклеротических заболеваний населения России

Кудрявцев В. Б., Расторгуев В. В., Рыжов А. П., Строгалов А. С.

Введение

Известно, что особенное внимание в России и в мире уделяется проблемам, связанным с атеросклеротическими заболеваниями, при которых холестерин, продукты жизнедеятельности клеток, жировые субстанции, кальций, фибрин «налипают» на стенки артерий, в результате чего артерии утолщаются и отвердевают. Таким образом ограничивается поток крови в артериях, нарушается гемодинамика, что в свою очередь, может вызывать, в частности, повышение артериального давления, возникновение тромбов и др.

Россия – страна с одним из самых высоких показателей смертности от сердечно-сосудистых заболеваний. При этом финансирования, которое получают ведущие медицинские центры страны, занимающиеся проблемой атеросклероза и его последствий явно недостаточно для борьбы с этим заболеванием.

Поэтому перед институтом профилактической медицины МЗ РФ встала естественная задача о разработке методик и об отслеживании на новом технологическом уровне факторов риска атеросклеротических заболеваний у населения России, прогнозирование дальнейшей динамики заболеваний этого класса в различных его возрастных группах и выработка наиболее экономически эффективных способов уменьшения риска.

В предыдущей статье авторов [10] был описан первый этап данной работы:

- 1. Была выбрана технология информационного мониторинга (ТИМ) как платформа для мониторинга. Выбор был сделан в пользу ТИМ, потому что в отличие от широко применяемых систем традиционного мониторинга (например, экологического, медицинского, технического и т.п.), базирующихся на анализе большого количества результатов показаний различных измерительных приборов, датчиков, накопленной статистики и др., приставка "информационный" подчеркивает, что для интересующих нас систем, "входом" является не только показания приборов исключительно в числовой форме, а и некоторая нечисловая информация (хотя могут использоваться и показания измерительных устройств). Иначе говоря, мы создаем информационные системы использующие не только результаты измерений технических устройств, но и качественные оценки человеком различной информации, представленной в виде текстов, событий, действий по достижению, например, каких-то целей и т.д. тем самым мы включаем человека, как регулярно действующий «измерительный прибор», в систему мониторинга.
- 2. Был выбран подход, использующий методы нечеткой логики [3], поскольку в рамках поставленной задачи специалисты часто используют нечеткие понятия, такие как, например, «высокое давление», «плохая наследственность» и т.д. Методы нечеткой логики позволяют формализовать подобные понятия.
- 3. Был описан план построения системы информационного мониторинга заболеваемости:
 - а. Построить структурную модель предметной области.
 - b. Построить поведенческую модель предметной области (модель правил).
 - с. Заполнить модель информацией.
- 4. Была построена структурная модель предметной области.

Следующий этап: построение модели правил

На следующем этапе построения системы информационного мониторинга атеросклеротических заболеваний россиян, возникает задача построения модели правил предметной области.

Дело в том, что при попытках реализации на основе ТИМ практических систем, разработчики часто сталкиваются с проблемами при построении моделей правил предметной области. И это иногда делает невозможным применение технологии для решения конкретной задачи. Проблема заключается в том, что зачастую трудно найти хороших экспертов в предметной области, но даже если такие эксперты найдены, то может возникнуть ситуация, когда хороший эксперт не может явно сформулировать свои знания; кроме того, обычно число всевозможных факторов в исследуемой предметной области настолько велико, что даже хороший эксперт может просто не заметить всех связей между ними.

Вообще говоря, описанная проблема встречается не только при построении систем информационного мониторинга, но и во многих других областях научной деятельности. Именно поэтому задача автоматического построения моделей поведения предметной области является очень актуальной и практически важной для развития современных интеллектуальных систем.

Одной из технологий, которая предназначена для решения этой задачи, является технология углубленного анализа данных data mining (извлечение знаний), которая осуществляет поиск неочевидных закономерностей в информации, описывающей конкретную предметную область.

Введение в технологию data mining

Технология data mining — это технология, предназначенная для обнаружения скрытых фактов и закономерностей в базах данных. Фактически data mining — это набор различных методов, таких как, например, нейронные сети, генетические алгоритмы, деревья решений, методы кластеризации и многие другие. Но, наверное, одним из наиболее интересных и практически полезных методов, являются ассоциативные правила. Ассоциативные правила выражают зависимости между различными полями таблицы базы данных.

Нечеткие ассоциативные правила

Каждое ассоциативное правило представляется в форме $R = (x \in A) \Rightarrow (y \in B)$ или, кратко, $R = A \Rightarrow B$, где x и y – некоторые поля таблицы базы данных, A и B – некоторые множества.

Довольно часто экспертам и аналитикам приходится оперировать такими нечеткими понятиями, как, например, «низкое давление». В этом случае применение *нечетких* ассоциативных правил дает им простой и естественный способ работы с ними. Поэтому в дальнейшем в этой работе мы будем рассматривать только нечеткие ассоциативные правила. Заметим, однако, что классические «четкие» правила являются лишь частным случаем нечетких.

Итак, пусть A и B – нечеткие множества, определенные на универсумах U и V соответственно.

Каждая таблица базы данных состоит из нескольких полей (столбцы) и набора табличных записей (строки). Введем следующие обозначения:

N – количество записей в базе данных

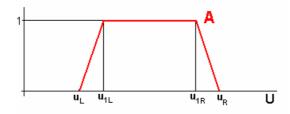
 $N_A(u)_-$ количество записей в базе данных, у которых значение признака из универсума множества A равно u.

 $^{\mathcal{X}_k}$ — значение некоторого признака в k-й записи (в правиле $A \Longrightarrow B$ этот признак из универсума множества A)

$$y(x_k)$$
 – значение признака из универсума множества B , соответствующее значению x_k

 $\mu_{A}(u)_{-3}$ начение функции принадлежности элемента универсума u нечеткому множеству A.

Будут рассматриваться трапецеидальные нечеткие множества, то есть множества, функция принадлежности которых представляет собой трапецию.



Функция принадлежности таких множеств имеет следующий вид:

$$\mu_{A}\big(u\big) = \begin{cases} \frac{u-u_{L}}{u_{1L}-u_{L}}, u_{L} \leq u \leq u_{1L}\\ 1, u_{1L} \leq u \leq u_{1R}\\ \frac{u_{R}-u}{u_{R}-u_{1R}}, u_{1R} \leq u \leq u_{R}\\ 0, \text{ иначе} \end{cases}$$

2.3. Характеристики нечетких ассоциативных правил

Опр. 3*начимость* (support) правила R:

$$\operatorname{supp} R := \frac{1}{N} \sum_{k=1}^{N} \mu_{A}(x_{k})$$

T.е. средняя степень принадлежности элементов базы понятию A. Это нечеткий аналог среднего числа элементов, принадлежащих A.

Опр. Уверенность (confidence) правила R:

conf
$$R := \frac{\frac{1}{N} \sum_{k=1}^{N} T(\mu_{A}(x_{k}), \mu_{B}(y(x_{k})))}{\frac{1}{N} \sum_{k=1}^{N} \mu_{A}(x_{k})}$$

где T — некоторая T-норма. T.е. нечеткий аналог доли тех элементов, у которых выполнены условия A и B среди элементов, в которых выполнено A.

Задача выбора оптимальных нечетких множеств

Для экспертов и аналитиков крайне желательным было бы извлечение правил с наивысшими значимостью и истинностью. Этого можно добиться путем правильного выбора нечетких множеств А и В. Например, при использовании методов нечеткой кластеризации для автоматического построения множеств А и В, мы можем произвольным образом варьировать их интервалы неопределенности. Каким же образом необходимо выбирать эти отрезки так, чтобы извлекаемые правила имели наивысшие характеристики? Ответ на этот вопрос зависимости между характеристиками правил и степенью нечеткости входящих в него множеств [9].

Оптимизация алгоритмов data mining для обработки нечетких понятий

Связь характеристик нечеткого ассоциативного правила и степени нечеткости входящих в него множеств

Авторами были доказаны следующие теоремы.

Теорема.

$$\operatorname{supp} R = \xi(A) + \frac{u_{1R} - u_{1L}}{|U|}$$

где $\xi(A)$ - степень нечеткости множества A.

Теорема.

$$\operatorname{supp} R \ge \xi(A)$$

 $\stackrel{-}{}_{arrho e} \xi(A)$ - степень нечеткости множества A.

 Π ри этом равенство достигается тогда и только тогда, когда A – треугольное множество.

Теорема. Пусть функция y(u) непрерывна на U. Тогда

$$\operatorname{conf} R \ge \mu_B \left(y \left(u^* \right) \right) \frac{\xi(A)}{\xi(A) + 1}$$

где $u^* \in U_-$ некоторая точка, μ_B - функция принадлежности нечеткого множества B.

Доказанные теоремы устанавливают зависимости между характеристиками нечетких ассоциативных правил и степенью нечеткости входящих в них нечетких множеств. Это позволяет выбирать нечеткие множества таким образом, чтобы полученные в результате ассоциативные правила представляли для экспертов наибольший интерес.

Стратегия извлечения нечетких ассоциативных правил

Рассмотрим нечеткое ассоциативное правило со сложными посылкой и следствием: $R=A_1\wedge\ldots\wedge A_n \Rightarrow B_1\wedge\ldots\wedge B_m=:A\Rightarrow B$

В общем случае, алгоритм перебора всевозможных комбинаций посылки и следствия – задача экспоненциальной сложности, поэтому на больших объемах данных и при большом количестве полей в базе данных, данная задача оказывается неподъемной даже для современных вычислительных систем.

Для того, чтобы сделать эту задачу решаемой, был разработан алгоритм стратегического (направленного) перебора нечетких ассоциативных правил, который, как показала практика, позволяет эффективно решать эту задачу даже на персональных компьютерах «среднего» уровня.

Итак, пусть $\Phi = \{F_1, \dots, F_n\}$ — набор полей таблицы базы данных, $A_{i1}, \dots, A_{ik_i}, i = 1 \dots n$ — набор нечетких множеств, определенных над полем Fi.

Тогда алгоритм полного перебора посылок нечетких ассоциативных правил можно сформулировать следующим образом.

- 1. **Рассматриваются посылки** длины **1.** Для каждого поля $F_i \in \Phi, i=1,\dots,n$ перебираем все множества $A_{ij}, j=1,\dots,k_i$, определенные над этим полем. В этом случае каждая посылка имеет вид $A=F_i \in A_{ij}$.
- 2. Рассматриваются посылки длины 2. Для каждого подмножества $F_{i_1}, F_{i_2} \subset \Phi$ перебираем все множества $A_{i_1j}, j=1,\dots,k_{i_1}$ и $A_{i_2j}, j=1,\dots,k_{i_2}$, определенные над полями F_{i_1} и F_{i_2} соответственно. В этом случае каждая посылка имеет вид $A=F_{i_1}\in A_{i_1j}\wedge F_{i_2}\in A_{i_2j}$
- к. Рассматриваются посылки длины к. В этом случае для каждого подмножества полей в Φ , состоящего из k элементов, перебираются все соответствующие определенные над ними множества. Каждая посылка имеет вид $A = F_{i_1} \in A_{i_1 j} \wedge \ldots \wedge F_{i_k} \in A_{i_k j}$. n-1.U так далее до шага n-1.

Точно такой же перебор используется для построения всевозможных следствий В нечетких ассоциативных правил. В этом случае, при переборе полей необходимо следить за тем, чтобы одно и то же поле не попало и в посылку, и в следствие. Тогда каждое ассоциативное правило очевидным образом получается путем «сцепления» посылки и следствия: $R = A \Longrightarrow B$, где A — перебираемые посылки, B — перебираемые следствия.

Авторами доказаны следующие утверждения.

Утверждение (антимонотонность значимости)

Пусть заданы два нечетких ассоциативных правила:

$$R = A \Rightarrow B, R' = A \land A' \Rightarrow B$$

Тогда

$$\operatorname{supp} R \ge \operatorname{supp} R'$$

Утверждение (антимонотонность истинности)

Пусть заданы два нечетких ассоциативных правила:

$$R = A \Rightarrow B, R' = A \Rightarrow B \wedge B'$$

Тогда:

 $conf R \ge conf R'$

Доказанные утверждения позволяют существенно сократить пространство поиска нечетких ассоциативных правил, заранее отсекая огромные ветви, проход по которым, в противном случае, занял бы существенное время.

Поскольку аналитикам и экспертам наиболее интересны нечеткие ассоциативные правила с по возможности наивысшими значимостью и истинностью, то на практике обычно задаются два барьера для этих характеристик: $0 < \varepsilon \le 1$ и $0 < \gamma \le 1$. Тогда технология data mining должна извлекать такие и только такие правила R, у которых $\sup R \ge \varepsilon$, conf $R \ge \gamma$. Эти барьеры аналитик варьирует по своему усмотрению, подбирая их в зависимости от конкретной задачи таким образом, чтобы в результате поиска были отобраны, только высокозначимые правила (то есть такие правила, которым можно доверять) и правила с достаточно высокой истинностью.

Таким образом, с учетом доказанных свойств антимонотонности, алгоритм стратегического перебора можно переформулировать следующим образом:

Если на k-м шаге алгоритма перебора для некоторого подмножества полей $\{F_{i_1},\dots,F_{i_k}\}\subset\Phi$ выполняется условие $\sup(F_{i_1}\in A_{i_1j}\wedge\dots\wedge F_{i_k}\in A_{i_kj}\Rightarrow B)<\varepsilon$, то на всех последующих шагах $p=k+1,\dots,n-1$ необходимо исключить из поиска все правила вида $F_{i_1}\in A_{i_1j}\wedge\dots\wedge F_{i_k}\in A_{i_kj}\wedge X\in A''\Rightarrow B$, где X – произвольное поле таблицы базы данных, A'' – произвольное множество над этим полем. Это следует из только что доказанного свойства антимонотонности значимости: значимость указанных правил, полученных на шагах $p=k+1,\dots,n-1$, будет не выше, чем значимость правил, полученных на шаге k, то есть меньше заданного барьера ϵ .

Аналогично для следствия, если на k-м шаге алгоритма перебора для некоторого подмножества $\{F_{i_1}, \dots, F_{i_k}\}$ $\subset \Phi$ выполняется условие $\sup(A \Rightarrow F_{i_1} \in A_{i_1j} \wedge \dots \wedge F_{i_k} \in A_{i_kj}) < \gamma$, то на всех последующих шагах $p = k+1, \dots, n-1$ необходимо исключить из поиска все правила вида $A \Rightarrow F_{i_1} \in A_{i_1j} \wedge \dots \wedge F_{i_k} \in A_{i_kj} \wedge X \in A''$, где X произвольное поле таблицы базы данных, A'' произвольное множество над этим полем. Это следует из только что доказанного свойства антимонотонности истинность указанных правил, полученных на шагах $p = k+1, \dots, n-1$, будет не выше, чем истинность правил, полученных на шаге X, то есть меньше заданного барьера X.

Заключение

Доказанные авторами утверждения позволяют решить распространенную в том числе в медицине проблему формализации знаний экспертов. Следующими шагами работы видится сбор данных по динамике атеросклеротических заболеваний в России и применение реализованных алгоритмов для построения модели правил предметной области.

Список литературы

- 1. Рыжов А. П. Оценка степени нечеткости и ее применение в системах искусственного интеллекта. Интеллектуальные системы. Т.1, Вып.1-4, Москва, МНЦ КИТ, 1996, с. 95 102.
- 2. Ryjov, A., Belenki, A., Hooper, R., Pouchkarev, V., Fattah, A. and Zadeh, L.A., Development of an Intelligent System for Monitoring and Evaluation of Peaceful Nuclear Activities (DISNA), IAEA, STR-310, Vienna, 1998, 122 p.
- 3. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приблизительных решений. М., Мир, 1976. 165 с.
- 4. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М., Наука, 1985 .- 320 с.
- 5. Саати Т. Анализ иерархических процессов. М., Радио и связь, 1993 315 с.
- 6. Messarovich M. D., Macko D., Takahara Y. Theory of hierarchical multilevel systems. Academic Press, N.Y.- London 1970 344 p.
- 7. Попов Э. В. Экспертные системы: решение неформальных задач в диалоге с ЭВМ. М., Наука, 1987. 288 с.
- 8. Usama M. Fayyad (Ed.). Advances in Knowledge Discovery and Data Mining. MIT Press, 1996, 560 p.
- 9. Рыжов А. П. Оценка степени нечеткости и ее применение в системах искусственного интеллекта. Интеллектуальные системы. Т.1, Вып.1-4, Москва, МНЦ КИТ, 1996, с. 95 102.
- 10. Ахмеджанов Н. М., Жукоцкий А. В., Оганов Р. Г., Кудрявцев В. Б., Расторгуев В. В., Рыжов А. П., Строгалов А. С. Информационный мониторинг в задаче прогнозирования риска развития сердечнососудистых заболеваний. // Интеллектуальные системы. Т.7, Вып.1-4, Москва, МНЦ КИТ, 2002-2003, с.5-38.

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ" (23-27 октября 2006 г.), том 1, часть 1. Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В., 2006 г., 172 с.

Подписано в печать 29.09.2006.

Формат 60×90 1/16. Объем 11.5 п.л.

Заказ 18. Тираж 200 экз.

Издательство ЦПИ при механико–математическом факультете МГУ г. Москва, Воробьевы горы.

Лицензия на издательскую деятельность ИД № 04059 от 20.02.2001 г.

Отпечатано на типографском оборудовании механико-математического факультета