Conference of PhD Students in Computer Science

Volume of extended abstracts

CS^2

Organized by the Institute of Informatics of the József Attila University

July 18-22, 1998
Szeged, Hungary

Scientific Committee:

Miklós Bartha
Tibor Csendes
János Csirik
József Dombi
Zoltán Ésik
Zoltán Fülöp
Ferenc Gécseg
Balázs Imreh
Attila Kuba
Eörs Máté
György Turán

Organizing Committee:

László Bernátsky
Tibor Csendes
Lajos Schrettner
Csongor Halmai
László Martonossy
Mariann Sebő

Address of the Organizing Committee

c/o. Tibor Csendes
József Attila University, Institute of Informatics
H-6701 Szeged, P.O. Box 652, Hungary
Phone: +36 62 454 305, Fax: +36 62 312 292
E-mail: cscs@inf.u-szeged.hu
URL: http://www.inf.u-szeged.hu/~cscs/

Sponsors

John von Neumann Computer Society
József Attila University
Ministry of Culture and Education
OMFB National Technological Development Committee
Szeged City Mayor's Office

Preface

This conference is a kind of an experiment: the organizers have tried to get together those PhD students who work on any fields of computer science and its applications to help them possibly in writing their first abstract and paper, and may be to give their first scientific talk. As far as we know, this is the first of its kind. The aims of the scientific meeting were determined on the council meeting of the Hungarian PhD Schools in Informatics: it should

- provide a forum for PhD students in computer science to discuss their ideas and research results,
- give a possibility to have constructive criticism before they present the results in professional conferences,
- to promote the publication of their results in the form of fully refereed journal articles, and finally
- to promote hopefully fruitful research collaboration between the participants.

To achieve these goals, each participant was asked to act as a discusser of a paper with similar subject to facilitate the discussion and to improve the quality of the emerging paper. The discussers were asked to contact the authors of the related paper well before the conference to get acquainted with the material to be prepared to ask questions and to make useful comments. The success of the discusser institution depends much on the amount of invested work of the participants: we hope that it proves to be a profitable task for each side.

A Best Talk award will be given for the highest quality paper presented, and the best papers of some subfields will be announced.

Although we did not advertise it on the web, a surprisingly high number of good quality abstracts have been submitted. If you encounter any problems during the meeting, please do not hesitate to contact one of the Organizing Committee members. The organizers hope that the conference will be a valuable contribution to the research of the participants, and wish a pleasant stay in Szeged.

Contents

Preliminary Program

Overview

Saturday, July 18

- 10:00 - 14:00 Registration
- 14:00 - 14:15 Opening
- 14:15 - 15:00 Plenary talk
- 15:00 - 15:15 Break
- 15:15 - 16:45 Talks in 2 streams (3x30 minutes)
- 16:45 - 17:00 Break
- 17:00 - 18:00 Talks in 2 streams (2x30 minutes)
- 18:30 - 20:30 Supper

Sunday, July 19

- 08:30 - 10:00 Talks in 2 streams (3x30 minutes)
- 10:00 - 10:15 Break
- 10:15 - 11:00 Plenary talk
- 11:00 - 11:15 Break
- 11:15 - 12:45 Talks in 2 streams (3x30 minutes)
- 12:45 - 14:00 Lunch
- 14:00 - 15:30 Talks in 2 streams (3x30 minutes)
- 15:30 - 15:45 Break
- 15:45 - 17:15 Talks in 2 streams (3x30 minutes)
- 18:00 - 19:30 Reception at the Town Hall

Monday, July 20

- 08:30 - 10:00 Talks in 2 streams (3x30 minutes)
- 10:00 - 10:15 Break
- 10:15 - 11:00 Plenary talk
- 11:00 - 11:15 Break
- 11:15 - 12:45 Talks in 2 streams (3x30 minutes)
- 12:45 - 14:00 Lunch
- 14:00 - 20:00 Excursion and supper

Tuesday, July 21

- 08:30 - 10:00 Talks in 2 streams (3x30 minutes)
- 10:00 - 10:15 Break
- 10:15 - 11:00 Plenary talk
- 11:00 - 11:15 Break
- 11:15 - 12:45 Talks in 2 streams (3x30 minutes)
- 12:45 - 14:00 Lunch
- 14:00 - 15:30 Talks in 2 streams (3x30 minutes)
- 15:45 - 16:00 Closing session, announcing the Best Talk Awards
- 18:00 - 20:30 Supper

Wednesday, July 22

- 8:30 Departure

Detailed program

Saturday, July 18

10:00	Registration	
14:00	Opening session	
14:15	**Plenary talk** Miroslav Ciric *Automata, algebras and semigroups*	
15:00	Break	
Sections	**Databases**	**Decision support systems, fuzzy**
15:15	Zoltán Kincses *Questions on complex dynamic security*	József Dombi and Nándor Vincze *Universal characterisation of non-transitive preferences*
15:45	István Szépkuti *Multidimensional or Relational. How to organize an On-Line Analytical Processing Database?*	Sasa Dimitrijević, Dragan Antić and Predrag Stanković *Fuzzy spark advance scheduling for internal combustion engine*
16:15	Antal Nagy, László Nyúl and Zoltán Alexin *Software development of Medical Image Archiving System*	Zsolt Kalmár *Module Based Reinforcement Learning for a real Robot*
16:45	Break	
Sections	**JAVA**	**Software engineering, networks**
17:00	Zsolt Werner *Using Java and Erlang in Protocol Testing*	Mazen Malek, Roland Geche *Combination of Conformance, Performance and Interoperability testing for Internet Applications*
17:30	Viatcheslav V. Ostapenko *Generation of JAVA classes from ASN.1 for BER transfer syntax*	Vladan Devedzić, Danijela Radović, Ljubomir Jerinić *The Components for Intelligent Tutoring Systems*

Sunday, July 19

Sections	**Operations research, optimization**	**Information systems, software engineering**
08:30	András Csallner and Mihály Csaba Markót *Improving Interval Methods for Global Optimization*	Károly István Boda *Complex Data Structures and their Role in the Organisation of Information Systems*
09:00	Gábor Péter Szabó and Leocadio G. Casado *Equal Circles Packing in the Unit Square*	Jouni Järvinen *Difference Functions of Dependence Spaces*
09:30	András Kocsor, László Tóth and Imre Bálint *On the optimal parameters of a sinusoidal representation of signals*	Gyula Tömösi *Hybrid Modelling and Reasoning in Measuring Systems*
10:00	Break	
10:15	**Plenary talk** Katalin Tarnay *Protocol development: software problems and solutions*	
11:00	Break	
Sections	**Optimization, grammars**	**Information systems, networks**
11:15	Gábor Magyar, Mika Johnsson, Olli Nevalainen *On the Exact Solution of the Euclidean Three-Matching Problem*	Slavoljub Milovanovic *Open information systems - serbian enterprises standing and perspectives*
11:45	János Balogh, Pilar Martinez-Ortigosa, I. García *Simulation and implementation of the Parallel Control Random Search Algorithm*	Attila Lakatos, Pál Tőke *On the partial correctness of the alternating hit protocol*
12:15	Judit Csima *On extended simple eco-grammar systems*	Tchanin Alexandr *Generation MSC and TTCN descriptions for ISUP supplementary services*
12:45	Lunch	

(see next page for the rest of the **Sunday** program)

Sunday, July 19 (continued)

Sections	**Automata theory**	**Networks and protocols**
14:00	Tamás Hornung, Sándor Vágvölgyi *Decomposition of CFT(S) transformations with look-ahead*	Adalla Khalil Areik *Rule-Based System for Conformance Testing*
14:30	Miklós Bartha, Miklós Krész *Elementary decomposition of soliton automata*	Mihály Bohus *Uniform test selection method*
15:00	Tatjana Petković *The Correspondence Between Varieties of Automata and Semigroups*	Endre Németh *Built-in scheduling for protocol design*
15:30	Break	
Sections	**Automata theory, numerical mathematics**	**Software engineering, networks**
15:45	Antal Pukler *On Length of Directing Words of Automata*	Dragan Janković *Construction of recursive algorithms for polarity matrices calculation in polynomial logical function representation*
Sections	**Numerical mathematics**	**Software engineering, networks**
16:15	Sasa Dimitrijević, Bratislav Danković, Dragan Antić *Simulation approach for localizing roots of real coefficients complex equations*	András Micsik *A study of portability in the deployment of WWW*
16:45	Gábor Kallós *The Structure of the Univoque Set*	Csaba Vilmos Rotter *Regular grammar model for protocol testing*

Monday, July 20

Sections	**Image processing**	**Software engineering, networks**
08:30	Attila Alföldi, Attila Bak, Richárd Gál, Tamás Szabad *Compression and Processing of Still Images Using Wavelet Transformation*	Miklós Berzsenyi *A Hierarchical Algoritm for Link-Sharing, Real-Time and Priority Services*
09:00	Attila Fazekas, András Hajdu *Analyzing the noise sensitivity of skeletonisation processes*	Zoran Putnik *An Analysis of Some Characteristics of Different Programming Paradigms*
09:30	Vesna Velićković *A filter to avoid the aliasing problem*	Zoltán Porkoláb *Using Object Oriented Techniques at Implementing Compilers*
10:00	Break	
10:15	**Plenary talk** György Maróti *Topics in Computer Algebra*	
11:00	Break	
Sections	**Image processing**	**Software engineering, networks**
11:15	Attila Tanács, Kálmán Palágyi *Efficient implementation of morphological and local neighbourhood operations*	Gabriella Kókai *Error Diagnosis in Prolog Programs, A Critical View*
11:45	Joonas Lehtinen *Limiting Distortion of a Wavelet Image Codec*	
12:15	Cs. Halmai, E. Sorantin, A. Tanacs, P. Winkler, G. Wolf *Investigations of Tubular Structures*	
12:45	Lunch	
14:00	Excursion	

Tuesday, July 21

Sections	**Discrete mathematics**	**Artificial intelligence**
08:30	Béla Csaba *On the Partitioning Algorithm*	Anikó Ekárt *Generating class descriptions of four bar linkages*
09:00	Ferenc Kruzslic *Improved Greedy Algorithm to Look for Median Strings*	Endre Fülöp *The n-distinguishable-queens problem, an extension of a classical AI problem*
09:30	József Békési, Gábor Galambos and Péter Hajnal *Analysis of permutation routing algorithms*	Ádám Schmideg *Automatically generated icons for documents of electronic libraries*
10:00	Break	
10:15	**Plenary talk** Péter Hajnal *Decision tree complexity - a graph theoretical approach*	
11:00	Break	
Sections	**Discrete mathematics**	**Artificial intelligence**
11:15	Csaba Holló, Zoltán Blázsik, Csanád Imreh, Zoltán Kovács *On a merging reduction of the Process Network Synthesis Problem*	Péter Olaszi, Ilona Koutny, Gábor Olaszi *Syntactical Analysis of Hungarian Sentences to Produce Prosodic Information for Speech Synthesis*
11:45	Harri Hakonen, Timo Raita *A Fast Constant-Space Substring Search Algorithm*	András Péter, Lehel Csató *A New Approach to Neural Network Design*
12:15	Dragan Stevanović *Constructing Graphs with Given Eigenvalues and Angles*	Miklós Hoffmann, Emőd Kovács *Training of Artificial Neural Networks by Linear Mappings*
12:45	Lunch	

(see next page for the rest of the **Tuesday** program)

Tuesday, July 21 (continued)

Sections	**Software engineering and applications**	**Artificial intelligence**
14:00	Jaakko Järvi *Object Oriented Model for Parameter Estimation of Partially Separable Functions*	Tommi Johtela, Jouni Smed, Mika Johnsson and Olli Nevalainen *Applying Fuzzy Multiple Criteria Optimization to PCB Scheduling*
14:30	Prof. Borko Kristić, Ognjen Radović, Srdjan Marinković, Ksenija Dencić *Some aspects of financial instruments' price modelling*	Ljubomir Jerinić, Vladan Devedzić *The Cost-Effective and Component Based Intelligent Tutoring Shell — the GET-BITS Model*
15:00	Predrag V. Krtolica *A New Look at Reverse Polish Notation*	Lehel Csató, András Péter *Neural Network Model for Nonlinearity Detection*
15:45	Closing session, announcing the Best Talk Awards	

Wednesday, July 22

- Departure

Compression and Processing of Still Images Using Wavelet Transformation

Attila Alföldi, Attila Bak, Richárd Gál, and Tamás Szabad

In the last decade there has been an enormous increase in the exchange and storage of information. Together with this growth, there has also been a change in the type of information: visual information in the form of digital images has become increasingly important. This type of information especially gives rise to high transmission and storage costs. Therefore the development of reliable and fast compression techniques for several quality levels has become an important research topic. Many algorithms have been proposed in the literature and some of them have already been standardized. However, standardization is a time consuming process and development goes on. There is a need for additional standards and most of all for special purpose algorithms. The compression technique we would like to introduce uses multiple resolution of the signal, and is based on the wavelet theory. Our work focuses on the adaptive uniform scalar quantization of a 64-subband discrete wavelet transform image decomposition (DWT), followed by zero run-length and Huffman coding. The official specification is referred to as the wavelet/scalar quantization (WSQ) standard. In the essay we intend to cover the following topics:

- Explanation of the wavelet theory from the viewpoint of signal processing. In the analysis of signals it is often useful to observe a signal in successive approximations. This can be best described by the multiresolution analysis of signals. We introduce the wavelet expansion of signals and the dyadic subband tree.
- The image coding scheme as used in our experiments. 2D-DWT, quantization, bit allocation, entropy coding.
- Comparison of different compression techniques with special emphasis on WSQ and DCT algorithms (JPEG).
- Further development possibilities of the DWT in the field of video compression. Fundamental video coding schemes.

In the future, the variety of communication systems will be merged into one system that handles all types of information (multimedia). There will no longer be a clear distinction between PC, telephone and television. For this system, storage and transmission of the information will be standardized and designed to be, as much as possible, independent of the application area. For further research we would like to investigate the possibilities of applying the wavelet transform in the field of video communications. In the second part of the essay we introduce new techniques for accelerating the decoding phase of DWT based compressed still images and fast FIR filtering of digital images given in (scalar quantized) DWT domain. The proposed new filtering techniques cover spatially invariant and variant 2D FIR cases. The proposed methods offer significant reduction in the computational costs of the decoding and filtering operations. The methods can be applied in DWT based video compression schemes.

Rule-Based System for Conformance Testing

Adalla Khalil Areik

My goal is to apply rule-based system for conformance testing. This paper will discuss the following:
- the basic for conformance testing,
- the test notation TTCN (Tree and Tabular Combined Notations),
- the fundamentals of rule-based system
- how the TTCN and the rule-based system can be related to each other.

Conformance testing is the assessment process to determine the extent to which an implementation of an OSI (Open System Interconnection) standard conforms to the requirements stated in that standard. Conformance testing has an abstract notation, called TTCN (Tree and Tabular Combined Notations). The TTCN is designed for expressing all attribute of an abstract test suite. The syntax and semantics of TTCN are both tightly coupled to the international standards for conformance testing. A conformance suite in TTCN consists of a number of test cases. One of the main parts deals with dynamic, containing the actual description of the test case.

I present the components of TTCN dynamic behavior tree as a set of rules. These components include Arcs which represent input and output, Nodes represent the determination of Arc and f (the relationship between the components). I develop a test case tree definition, a path matrix and rules for test cases, and I attempt to introduce new measures for uncertainty in conformance testing, threshold level and confidence factor of the output for any rule.

Finally the conversion of the dynamic behavior tree to rule-based system is explained. The above listened results are demonstrated on the INRES protocol.

An Intelligent Cooperative Information Retrieval System Based on Multi-Agency

Mihal Badjonski

An agent can be defined as a hardware or (more usually) software based computer system that enjoys the following properties:

* autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;

* social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;

* reactivity: agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;

* pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.

A system compound of two or more agents is a multi-agent system.

A network of inter-operating information sources is often referred to as intelligent and cooperative information system.

This paper presents an intelligent cooperative information system, which is developed as a multi-agent system. User communicates with an agent (no matter which one) from the system using a web browser. User may add some article to the system, request for the particular article from the system or it may submit a query for the intelligent search of the articles. When adding a new article, meta information about the article have to be specified (keywords, author name, date, type of the article, etc.). These meta information will be used for intelligent search when a user asks for the list of the articles that best matches his/her specification. The system methodically distributes the articles over its agents, in order to facilitate the search. The system is extensible, new agents can be easily added as it grows.

The most attractive features of this information system are:

* multi-agent nature - there is no central part of the system that controls its overall performance. Its intelligence and control strategy emerges from agents cooperation and local policy.

* parallel distributed search - as opposed to less efficient centralized search done by WWW searching machine,

* distributed information acquisition - everyone can add articles (like in WWW)

* extensibility - from the prototype version it may grow into global system (such as WWW),

* robustness - if some agents are out of order the system continues its work with graceful degradation,

* platform independence - the system is implemented in Java 1.1 which is available for most operating systems.

The system resembles to a digital encyclopedia (such is The Britannica) and the World Wide Web. The Java classes used in the system implementation utilize the Internet for agent to agent communication. This means that system can be distributed all over the globe.

Simulation and implementation of the Parallel Control Random Search Algorithm[1]

J. Balogh, P.M. Ortigosa, and I. García

In this work we propose to study the Controlled Random Search (CRS) algorithm of Price [1, 2] in order to get an efficient parallel version. Some parallel approaches have been proposed by García [3, 4], McKeown [5], Sutti [6], Ducksbury [7], Price [8] and Woodhams and Price [9] using several kind of parallel computers and strategies.

The Price sequential procedure to globally minimize $f : R^n \to R$ can be summarized as follows:

1. Generate N random points $R_0, \ldots, R_{N-1}$ in the feasible region (F.R.) and evaluate f in each point.
2. Find the best point $(R_B, f(R_B))$ and the worst one $(R_W, f(R_W))$ among $x_1, \ldots, x_N$, such that $f(R_B) \leq f(R_k)$, and $f(R_W) \geq f(R_k)$, for all k.
3. Choose randomly $n+1$ points $R^0, \ldots, R^n$. Compute $\overline{P} = 2 \times \overline{G} - \overline{R}^n$, where $\overline{G}$ is the centroid for $R_0, \ldots, R_{n-1}$. If $\overline{P}$ is in the F.R and $f(\overline{P}) \leq f(R_W)$, substitute R_W, with $\overline{P}$in the sample set, otherwise if success rate $> 50\%$, compute $\overline{P} = \frac{\overline{G}+\overline{R}_n}{2}$. If $f(\overline{P}) \leq f(R_W)$ holds for the point $\overline{P}$, then substitute R_W, otherwise repeat step (iii),
4. Stop if the termination criterion is satisfied, otherwise go to step (ii).

In the above scheme, let us distinguish three stages: The first of Initialization (I.S.) consisting of step (i), which occurs once only in the procedure, the second of Update (U.S.) consisting of step (ii) and the third of Exploration (E.S.) consisting of steps (ii) and (iv).

As it can be seen, this algorithm is highly sequential, because each new point $\overline{P}$ is generated from the sample set which consist of the best previous points computed before it. However, it is possible to detect some parallelism. At the initialization stage $R_0, \ldots, R_{N-1}$ can be simultaneously generated, and $f(R_0), \ldots, f(R_{N-1})$ can be simultaneously computed.

So it seems that the best strategy to parallelize this algorithm is the global one, where a master-worker communication model is required. In this strategy the master processor executes the PCRS algorithm and a worker processor works only with the trial points supplied by the master processor. In our strategy the master processor generates the points and sends them to the worker processors. In this way, a worker processor only evaluates the objective function at the trial points supplied by the master processor. After every evaluation it sends the result back to the master [4].

García et al. [10, 4] implemented this strategy. Their algorithm was fully asynchronous except at the end of the initialization stage, where the master did not start to generate $\overline{P}$ points until the initial sample set was evaluated. Once the set was initialized, the master processor generated NP (Number of Processors) trials points and sent them to the workers, so each worker processor received a trial point to evaluate. After every evaluation, workers send the result back to the master and receive a new point to evaluate. Although this approach is asynchronous, it is possible that a worker processor has to wait for a new point if another worker(s) has (have) sent its (their) function evaluation to the master at the same time. It could be solved if the worker processors have more than one trial point to evaluate. So, when a worker processor finishes an evaluation and sends the result to the master, it can go on evaluating another of its points while the master receives the information and sends a new point back to the worker processor. In this way not only the above problem is solved, but also the overhead from the communication cost is reduced, because it is overlapped with some computation cost.

However, another problem arises of this kind of asynchronous parallel versions. Data, which are used in simultaneous processes may belong to different iterative cycles, so the master processor generates new

[1]This work was supported by the Tempus JEP 11264-96, by the Consejeria de Educación de la Junta de Andalucía (07/FSC/MDM) and by the Ministry of Education of Spain (CICYTTIC96-1125-C03-03).

trial points from the sample set which has not all the information about the most recently generated point. This may produce that the new points will not be as good as in the sequential case, and then more trial points have to be generated, and the number of evaluations in the parallel version can be higher than for the sequential one. This problem raised in the García's implementation where the worker processors only had a point to evaluate each time. This increase in the number of function evaluations can even be higher in the case when the worker processors have two or more points to evaluate.

Therefore, it is rather important to know the effects of increasing the number of points that each worker processor stores to send sequencialy to the master processor once one of this stored points is evaluated.

We are changing the sequential version of CRS in order to simulate the behaviour of the new parallel version where a worker processor has some points to evaluate instead of only one. We will study the effects of these changes and determine the amount of trial points that a worker processor must have to get the best performance.

We will implement the parallel version (previously simulated and studied) in a MIMD system using PVM and MPI, and its performance will be analyzed.

A set of test functions will be used to check the convergence and the parallel performance. As this algorithm has a strong stochastic component, the number of function evaluations needed to reach the global minimum depends on the particular execution. For this reason the algorithm will be executed 100 times for each case obtaining a stable statistical sample. From this data set, mean value (μ) of the number of function evaluations and the corresponding confidence intervals (95%) will be computed (see [11]). So, the probability of an interval covering the mean is 0.95 or, expressed in another way, that on the average 95 out of 100 confidence intervals similarly obtained would cover the mean.

References

[1] W.L. Price, "A controlled random search procedure for global optimization," in *Towards Global Optimization 2* (L.C.W Dixon and G.P. Szegö, ed.), pp. 71–84, Amsterdam: North Holland, 1978.

[2] W.L. Price, "Global optimization algorithms by controlled random search," *Journal of Optimisation Theory and Applications*, no. 40, pp. 333–348, 1983.

[3] I. García, G.T. Herman, "Global optimization by parallel constrained biased random search," in *State of Art in Global Optimization: Computational Methods and Applications* (C.A. Floudas and P.M. Pardalos, eds.), Kluwer, Dordrecht, pp. 433-455.

[4] I. García, P.M. Ortigosa, L.G. Casado, G.T.Herman and S. Matej, "Multidimensional Optimization in Image Reconstruction from Projections," *Developments in Global Optimization*, Kluwer, Dordrecht, pp. 289–300, 1997.

[5] J.J. McKeown, "Aspects of parallel computations in numerical optimization," in *Numerical Techniques for Stochastic Systems* (F. Archetti and M. Cugiani, eds.), pp. 297–327, 1980.

[6] C. Sutti, "Local and global optimization by parallel algorithms for MIMD systems," *Annals of Operations Research*, no. 1, pp. 151–164, 1984.

[7] P.G. Duckbury, *Parallel Array Processing*. Chichester: Ellis Horwood, 1986.

[8] W.L. Price, "Global optimization algorithms for a CAD workstation," *Journal of Optimisation Theory and Applications*, no. 55, pp. 133–146, 1987.

[9] F.W.D. Woodhams, W.L. Price, "Optimizing accelerator for CAD workstation," *IEE Proceedings Part E*, vol. 135, no. 4, pp. 214–221, 1988.

[10] I. García, P.M. Ortigosa, L.G. Casado, G.T.Herman and S. Matej, “A parallel implementation of the controlled random search algorithm to optimize an algorithm for reconstruction from projections,” in *IIIrd Workshop on Global Optimization*, (Szeged, Hungary), pp. 28–32, December 1995.

[11] R.R. Sokal and F.J. Rohlf, *Biometry*. New York: Freeman, 1981.

Elementary Decomposition of Soliton Automata

Miklós Bartha and Miklós Krész

A soliton automaton is the mathematical model of so called soliton valves in certain carbohydrate molecules, and has the potential to serve as a future molecular switching device. The underlying object of a soliton automaton is a soliton graph, which is an undirected graph G having a perfect internal matching, i.e., a matching that covers all the vertices of G with degree at least two. Such vertices are called internal, whereas vertices with degree one are called external in G. The states of the automaton are all the perfect internal matchings of G, and transitions are defined by making alternating walks connecting two external vertices in G.

The main contribution of this paper is a decomposition of soliton automata according to the global internal structure of their underlying soliton graphs. For practical reasons, all edges of soliton graphs are assumed to be accessible by an external alternating path, i.e., alternating path starting from an external vertex. The elementary components – maximal connected subgraphs spanned by allowed edges – of G can be grouped into disjoint families in a way that:

1. Each family has a unique gate member through which all other members can be accessed by external alternating paths.
2. For each family, all external alternating paths reaching that family arrive in vertices belonging to the same canonical class of the gate.
3. Each family has an ear decomposition by which all non-gate members are located on an appropriate thread around the gate.

The families above can be arranged in a fixed partial order reflecting the order in which they are accessible by external alternating paths. The corresponding soliton automaton, too, can be decomposed into component automata determined by these families. For each internal family F, the component automaton defined by F will have a transition between any two of its different states. A transition from any particular state to itself, however, will exist if and only if the family represents a non-bipartite subgraph.

Analysis of permutation routing algorithms[2]

József Békési, Gábor Galambos, and Péter Hajnal

In this paper we analyze some permutation routing algorithms for different kind of mesh architectures. We give lower bounds for the number of steps of arbitrary on–line or off–line algorithms on rectangular meshes with buses . Finally we give lower and upper bounds for the expected number of steps of the basic greedy algorithm on linear array without bus.

One possible alternative of the traditional single processor computer is the multiprocessor parallel computer. The effectiveness of such a computer highly depends on how fast the necessary data can be sent to the appropriate location. The architecture or an efficient routing algorithm can significantly decrease the developing costs of such a computer. Therefore researchers turned their attention to either the hardware architecture or the analysis of the different algorithms.

During the analysis different architectures used to be considered. One of the simplest architecture is the linear array. In this case each processor is connected with the two neighboring processors.

In some cases using buses is practical for data communication to increase the efficiency of the computer. The bus is connected to all processors, but only one processor may use the bus at a given step.

In this paper we deal with the permutation routing problem: each processor should send a message to another one. We deal with the classical variant, where each processor can receive at most one message. Suppose that the neighboring processors are connected with a full–duplex line, which enables them to exchange a message chosen from the message queue at each step. We say that the problem is solved, if each message arrived at its destination. In the simplest cases we assume that the processors have enough memory to store the waiting messages. The efficiency of the algorithm is measured by the number of steps required to solve the problem. In each steps only the neighboring processors can send messages to each other, or exactly one processor can use the buses to which it is connected.

The efficiency of the algorithms can be measured two ways: we can investigate the number of steps in the worst–case or we can analyse the expected number of steps for solving a problem. Formally we can give the above mentioned efficiency by the following: let $\pi = (i_1, i_2, \ldots, i_n)$ an example, in which the processor j send a message to the processor i_j $(j = 1, 2, \ldots, n)$. Denote Π_n the set of all permutations of n. If $S_A(n, \pi)$ denotes the number of steps of algorithm A for π in case of n processors, then the algorithm requires

$$W_A(n) = \max_{\pi \in \Pi_n} S_A(n, \pi)$$

steps in the worst–case.

Lower bound for two–dimensional bused meshes. Cheung and Lau proved that for an $n \times n$ bused mesh each algorithm A requires at least $0.691n$ steps. In this paper we investigate, whether this bound can be improved for a $m \times n$ rectangular mesh.

Theorem 0.1 *For each algorithm A on an $m \times n$ $(m \geq n)$ rectangular mesh with row and column buses*

$$W_A(m, n) \geq \max\left(\frac{2}{3}m, \frac{7n + 3m}{8} - \frac{\sqrt{(3m - n)^2 + 16n^2}}{8}\right)$$

It is easy to see that if $m = n$ then the above bound gives the Cheung és Lau [2] result for square meshes.

[2]This paper has been supported by the OTKA grant number T-016349 of Hungarian Academy of Sciences

Average case analysis for one dimensional routing problem. Consider now the one dimensional problem without bus. Denote n the number of processors, and let A be the basic greedy algorithm, which solves the problem such a way, that each processor sends its message to its neighbor in the direction of the destination processor. If the message arrives at its destination, then the processor stores the data, but it still remains able to receive and send further messages.

Then the number of steps required to solve the problem equals to the maximal distance of the messages from their destination. Formally we can say that if $\Pi = \pi(1), \pi(2), ..., \pi(n)$ defines the problem, then the number of steps required by algorithm A is

$$L_A^{\Pi}(n) = \max_{i=1,...,n} \{|i - \pi(i)|\}.$$

In the following we would like to determine the average–case behaviour of algorithm A. To do this we suppose that the algorithm gets each permutation of the elements $1, ..., n$ with equal probability. Denote $S_j(n)$ the number of permutations of elements $1, ..., n$ for which

$$\max_{i=1,...,n} \{|i - \pi(i)|\} = j.$$

Then the expected number of steps for A is

$$E_A(n) = \frac{\sum_{j=1}^{n-1} j \cdot S_j(n)}{n!}.$$

Because the values of $S_j(n)$ are not known exactly for each j, or they can be given by only a very complicated formula (see [8]), so our aim is to give a useful lower bound for this expected value. In the following we give the theorem for the expected number of steps of algorithm A.

Theorem 0.2 $E_A(n) \geq n + 2 - 2\sqrt{n+1}$.

From the above theorem the following surprising fact follows:

$$\lim_{n\to\infty} \frac{E_A(n)}{n} = 1,$$

i.e. the asymptotic average–case behaviour of algorithm A is not better than its asymptotic worst–case behaviour.

References

[1] A. Bar-Noy and D. Peleg: Square Meshes are not always Optimal, IEEE Transactions on Computers, (1991) Vol 40, No. 2, 196-204.

[2] S. Cheung and F.C.M. Lau: A lower bound for permutation routing on two-dimensional bused meshes, Information Processing Letters 45(1993) 225-228.

[3] M. Kaufmann, R. Raman, J.F. Sibeyn: Routing on Meshes with Buses, Algorithmica 18(1997) 417-444.

[4] T. Leighton, F. Makedon és I.G. Tollis: A $2n - 2$ step algorithm for routing in an $n \times n$ array with constant size queues, Algorithmica (1995) 14: 291-304.

[5] T. Leighton: Average–case analysis of greedy routing algorithms, Proc. 2nd ACM Symposium on Parallel Algorithms and Architectures, 1990, 2-8

[6] J.Y.T. Leung and S.M. Shende: Packet routing on square meshes with row and column buses, Proc. 3rd IEEE Symp. on Parallel and Distributed Processing (1991) 834-837.

[7] J.Y.T. Leung and S.M. Shende: On multi-dimensional packet routing for meshes with buses, Tech. Rept. TR-150, Dept. of Computer Science and Engineering, University of Nebraska at Lincoln, 1992.

[8] L. Lovász: Combinatorial Problems and Exercises, Akadémiai Kiadó, Budapest, 1979.

Using JAVA to Provide a Reliable Distributed WEB

András Benczúr and Balázs Csizmazia

Large computer networks like the Internet are heterogeneous with many diverse hardware components running a more diverse spectrum of system software and user-level applications. It is partly because many organizations make their decisions on the basis of the costs of the full solutions of their problems without giving much attention on factors e.g. who provides the solutions or what kind of references has the vendor providing the solution. Another cause of this heterogeneity is the fact that already purchased systems may be too costly to replace, and so they are not likely to be replaced even if they are not based on up-to-date technologies. Of course this heterogeneity has its advantages, too: it allows to utilize the best technologies for each given problem field. Other characteristics of such large networks are that the underlying network protocols (like the TCP/IP, OSI families of protocols) are unreliable, and they cannot tolerate any partial or total failures of the utilized network media. Nowadays there are several distributed applications and there are even more applications under construction needing fault-tolerance, so that these applications can be relied upon in mission-critical settings too, even if they are built on non fault-tolerant networking technologies. Such are application areas e.g. for medicine, aircraft control, or trading applications built entirely on Internet technologies, and needing predictable behavior despite system failures (we have in mind here mainly TCP/IP-based network technologies with so-called halting failures where a component simply stops working, or a network link breaks between active network components). A nice example of this kind of setting is the WWW, which is used worldwide for accessing information and utilizes many components that can fail independently. Another WWW-related important concept is Java a new tool of distributed application development for the WWW by making the client side of the WEB more interactive by a code written in a portable high-level programming language. But it lacks a distributed object infrastructure, like the Arjuna or the COSS/TS service.

In this paper we describe a new tool providing a reusable reliable distributed infrastructure upon which reliable web-applications can be built easily. The tool is constructed in an object-oriented manner allowing an application to bracket a series of method invocations by begin/end transaction markers. If some of the method invocations fail during a transaction, the transaction manager tool will roll it back by undoing the effects of all previous method calls of the transaction. This tool is implemented in the Java language and allows an easy integration of the distributed Java objects implementing a transaction participation marker interface (also non-Java objects can be involved in transactions by providing a simple Java wrapper or other gateways).

Our tool uses the generally available Java remote object access facilities leveraging the programmer from the error-prone task of writing communication codes. In this paper we present its implementation utilizing distributed transaction processing protocols, and compare two well-known distributed transaction manager protocols, the 2PC and 3PC ones. Although 3PC is a non-blocking protocol, it doesn´t tolerate network link break failures typical for present day Internet technologies. This non-blocking protocol is more complex to program (the complexity arises from the fact that the 3PC protocol requires a fault-tolerant coordinator election protocol whose implementation is not straightforward on asynchronous network technologies, like the Internet). The tool provides, among the transaction manager service, a simple concurrency control service that can be used to implement distributed resource locking (also missing from the Java programmers tools).

Finally we are going to discuss the 3PC protocol´s applicability in faulty asynchronous networks, and to identify some areas where its added complexity can be tolerated, because of its additional non-blocking capabilities, and we will compare it with other possible solutions to this problem, focusing on their applicability on the WEB.

A Hierarchical Algorithm for Link-Sharing, Real-Time and Priority Services

Miklós Berzsényi

The emerging integrated services networks will support applications with diverse performance objectives and traffic characteristics. While most of the previous research on integrated services networks has focused on guaranteeing QoS, especially real-time requirements, for each individual session, several recent work has argued that it is also important to support hierarchical link-sharing service.

With hierarchical link-sharing, there is a class hierarchy associated with each link that specifies the resource allocation policy for the link. A class represents some aggregate of traffic streams that are grouped according to administrative affiliation, protocol, traffic type, or other criteria.

In this paper, we study hierarchical resource management models and algorithms that support both link-sharing and guaranteed real-time services with decoupled delay (priority) and bandwidth allocation. We extend the service curve based QoS model, which defines both delay and bandwidth requirements of a class, to include fairness, which is important for integration of real-time and hierarchical link sharing services. The resulting Fair Service Curve link-sharing model formalizes the goals of link-sharing and real-time services and exposes the fundamental tradeoffs between these goals. In particular, with decoupled delay and bandwidth allocation, it is impossible to simultaneously provide guaranteed real-time service and achieve perfect link-sharing. We propose a novel scheduling algorithm called Hierarchical Approximating Service Curve (H-ASC) that approximates the model closely and efficiently. The algorithm always guarantees the performance for leaf classes, thus ensures real-time services, while minimizing the discrepancy between the actual service provided to the interior classes and the services defined by the Fair Service Curve link-sharing model. We implemented the H-ASC scheduler. By performing simulation and measurement experiments, we evaluate the link-sharing and real-time performances of H-ASC, and determine the computation overhead.

Complex Data Structures and their Role in the Organisation of Information Systems

Károly I. Boda

It is widely known that the accumulated knowledge of mankind rapidly grows, at a rate that often seems to be very hard to handle. Those who use the Internet, and especially the Web with its hypermedia capabilities, have to cope day by day with more or less difficulties in finding the relevant, or at least the adequate information they need. As a consequence, the various ways of organising the knowledge stored in different computers and networks are, with no doubt, of great importance. The ultimate aim of these efforts is to increase the effectiveness and efficiency of information retrieval (its relevance, completeness, etc.); the issue is, however, discussed only to a relatively small extent and chiefly in general for its importance in recent publications, compared to the amount of information available on other, mainly technical, questions of the Internet and the Web, including multimedia, programming of the Web, etc.

Setting out from some basic principles of information systems which have been used widely and proved effective long since, in this presentation a general and abstract model is discussed in detail as well as its applications for comparing various information systems with each other, and for establishing their efficiency, or stage of development. The *Multi-layer Architecture of Information Systems* (MAIS) model describes information systems as complex structures built of four layers. The function of layers can be generally outlined as forming abstract objects and object structures in different levels of knowledge in order to describe, or reflect different parts of an application, emphasising different characteristics of reality.

The four layer of the MAIS model are as follows:

- index layer;
- logical/conceptual layer;
- textual layer;
- hipertextual layer.

In representing the content of information systems, the abstraction level (e.g. concepts, abstract or syntactic objects, models, semantic schemes, and paradigms, roughly) and complexity of layers are different. The layers are complex structures, the components of which are related to each other in horizontal level (i.e. within layers), and vertical level (i.e. across layers).

The function of layers can be outlined as follows:

- the index layer identifies the attributes, or characteristics of abstract objects described by the information system;
- the logical/conceptual layer identifies the abstract objects and their high-level structures, described by the information system;
- the textual layer contains concrete representations of abstract objects and their structures, described by the information system;
- the hipertextual layer implements various links between segments of the textual layer, organising their content in high level; besides, the links and series of links establish contacts with the index layer and logical/conceptual layer, respectively.

Examining the layer structure of special, well-tried information systems, the detailed functions of each layer can be expressed. In this presentation the realisation of layers are reviewed as well as their relation with complex data structures applied in the information systems listed below. The information systems, as well as other means and methods of organising knowledge in traditional and modern way (i.e. by computers), which have been taken into consideration in developing the MAIS model, are as follows:

- traditional, "linear" texts, i.e. books, articles, reports, etc.; their structure and organisation, as well as various methods for analysing and processing them in computer linguistics, content analysis, and literary texts analysis; bibliographies, word frequency statistics, concordances, etc.;
- encyclopedias, lexicons, mono- and bilingual, "active study", etc. dictionaries, thesauruses;

- library information systems, bibliographic and full text information retrieval systems;
- relational database systems;
- multimedia dictionaries and encyclopaedias (e.g. ENCARTA, BOOKSHELF);
- expert systems;
- hipertext/hypermedia systems, the World Wide Web itself.

As an application of the consequences drawn from the construction and study of MAIS model, in the presentation the organisation level of World Wide Web, and some future trends are outlined. The examples discussed above as well as presentations of interactive systems (e.g. through the Web, if the need arises) are intended to illustrate and back up significant parts of the above considerations.

Uniform Test Selection Method

Mihály Bohus

In protocol testing to avoid the complexity of "total" testing is common to select the "appropriate" test cases.

The test cases are derived from the formal descriptions (automatic selection) or from the protocol tester specialists (explicit selection). They apply the well-known test selection methods for functional, control and data-flow testing.

We propose the uniform selection method which exhibits the following properties :

- it can combine the experiences of the various formal description technics (Estelle, LOTOS and SDL FDTs),
- it can formalise the testing purposes and give some measure of the testing activity,
- it can apply in more phases of the protocol "life-cycle" namely in the design of the formal specifications (giving the testing information in the specification), in the selective derivation methods from the predefined formal specifications, in the selecting of the predefined test suites for the conformance and interoperability testing,
- it can analyze the data unit dependence to select the shortest test suite.

On the Partitioning Algorithm

Béla Csaba

The *paging problem* is defined as follows. We have a two-level memory system with k pages of fast memory, and $n-k$ pages of slow memory. Repeatedly a request to a page appears. This request should be satisfied by moving the page to fast memory, if it is in slow memory, i.e., a page fault occurs. In this case a page must be evicted from fast memory to make room for the new, recently requested one. The paging problem is to decide which page is to be evicted.

There is a simple optimum paging algorithm, called MIN, if we know the whole request sequence in advance, in the *off-line* case. It is more practical to consider the *on-line* paging, when the algorithm has to decide immediately after a page request, without knowing what the future requests will be.

For comparing two paging algorithms the *competitive ratio* is used. This measure of performance of an on-line algorithm was introduced by Sleator and Tarjan (see [ST]). Fix any starting configuration of the pages, and denote by $opt(\sigma)$ the optimum number of page faults on request sequence σ, in other words, the optimum cost of σ. The competitive ratio of the on-line algorithm $\mathcal{A}$ is c, if there is a constant M such that on every request sequence σ the cost incurred by $\mathcal{A}$, $\mathcal{A}(\sigma)$ is at most $c\,opt(\sigma) + M$. It was shown (see [ST]) that no on-line algorithm can have a competitive ratio less than k. $LRU, FIFO$ and a large number of other on-line algorithms are known to be k-competitive.

As it happens frequently, one may expect a better performance in the randomized case. A randomized on-line algorithm $\mathcal{R}$ is c-competitive, if there is a constant M such that on every request sequence σ $E[\mathcal{R}(\sigma)]$ is at most $c\,opt(\sigma) + M$, where $E[\mathcal{R}(\sigma)]$ denotes the expected cost incurred by $\mathcal{R}$ on σ. It was proved (see [FKLMSY]) that $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$ is a lower bound for the randomized competitiveness of an on-line paging algorithm. There is a simple, elegant algorithm, which has randomized competitive ratio $2\,H_k$. On the other hand, the only known optimal randomized algorithm, the *partitioning algorithm* has a much more complicated description.

One of our results is that we give another description of the partitioning algorithm via another approach. This is done by analyzing the optimal satisfactions of request sequences. We construct a graph, and the random walks on this graph correspond to the random choices done by the partitioning algorithm. Besides, we prove that the algorithm is optimal, k-competitive in the deterministic case, and $(k-l)$-competitive having an *l-strong lookahead*, where $k - l > 1$. This means that the on-line algorithm not only knows the present request and the previous ones, but has access to the first l element set of the future requests. No on-line algorithm can have less competitive ratio than $k-l$ with l-strong lookahead (see [A]), thus, the partitioning algorithm is optimal in this sense, too.

References

[A] Albers, S., *The influence of lookahead in competitive paging algorithms*, Proceedings of the 1st Annual European Symposium on Algorithms, 1993, pp. 1-12

[FKLMSY] Fiat, A., Karp, R., Luby, M., McGeoch, L., Sleator, D., Young, N., *Competitive paging algorithms*, 1991, Journal of Algorithms, 12:685-699

[ST] Sleator, D. D., Tarjan, R. E., *Amortized efficiency of list update and paging rules*, Comm. of the ACM, February 1985, pp. 202-208

Improving Interval Methods for Global Optimization[3]

A.E. Csallner and M. Cs. Markót

The global optimization problem can be defined in general as follows:

$$\min_{x \in X} f(x) \tag{1}$$

where X is a — possibly multidimensional — interval. If we denote the set of real intervals by $\mathbb{I}$ and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function of the problem, then $X \in \mathbb{I}^n$. Note, that a great class of real-life bound-constrained global optimization problems are covered by (1), e.g., problems where the parameters are given with tolerances or if the optimizers are supposed to be inside a parameter region [2].

Problem (1) can be solved with verified accuracy with the aid of interval methods (see, e.g., [1, 3, 4, 5, 6]). These methods are based on the well-known branch-and-bound principle. Thus, a search tree is built where the whole search region — the interval X — is the root and the particular levels consist of subintervals which are partitions of their parents in the tree. Those branches that cannot be pruned have to be stored for later treatment. The kind of the storage method used can be of great importance in performance if the increase of the number of intervals to be stored is considerable. The current presentation deals with these possibilities, i.e., the list handling of interval methods for global optimization.

Because the efficiency of a branch-and-bound method depends highly on which branch, i.e., stored list element is chosen as next to be treated, two basically different principles can be applied to handle the list. The first is to keep the list ordered and always pick up the first (or last) element, the second is to let the list be unordered and search for the next element in each step a new one is needed. The former saves computational time at picking up the elements, the latter at putting them onto the list. The time necessary for list handling can be calculated in both cases.

There are different principles to realize interval subdivision methods independently from the list handling. These variations can influence also the choice of the kind of list handling. There exist algorithms where the natural ordering of the list is the proper ordering for choosing the new elements to be picked up, hence, the list becomes a FIFO list.

A further possibility is to keep a part consisting of a constant number of list elements ordered, and refresh this part from time to time, e.g., after each list operation or when the ordered part becomes empty.

The oral presentation discusses all possibilities mentioned above thoroughly.

References

[1] Alefeld G. and Herzberger J. (1983), *Introduction to Interval Computations*, Academic Press, New York.

[2] Csallner A.E. (1993), *Global Optimization in Separation Network Synthesis*, Hungarian Journal of Industrial Chemistry, 21, pp. 303–308.

[3] Hansen E.R. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

[4] Kearfott R.B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht.

[5] Moore, R.E. (1966), *Interval Analysis*, Prentice Hall, Englewood Cliffs NJ.

[6] Ratschek H. and Rokne J. (1993), *Interval Methods*, In: Handbook of Global Optimization, Horst R. and Pardalos P.M. (eds.), Kluwer, Dordrecht, pp. 751–828.

[3]This work has been supported by the Grants OTKA T017241, F025743, and FKFP 0739/1997.

Neural Network Model for Nonlinearity Detection

Lehel Csató and Péter András

A neural network based algorithm to detect nonlinearities in a specified timeseries is presented. This method tests whether a multilayered neural network can detect nonlinearities in the application domain where it is used (sample data from the future application are given). The use of this test is useful in domains that require huge amounts of computation (in the applications the neural networks are equipped with many hidden neurons). If the neural network gives no significant contribution to the system output, then by eliminating the nonlinear part, a comparative and less "expensive" architecture is obtained.

The aim of this model is twofold: first to give a method to decide which model to use when implementing a forecasting algorithm, and second to combine the linear and nonlinear techniques in order to achieve a better prediction performance.

The network is built up using linear approximators (direct linear input-output connections in neural network terminology) and a nonlinear part. The neural network has the following form:

$$y = A \cdot x + R \cdot f(Q \cdot x) \tag{1}$$

where A is the autoregressive term; Q and R specify the nonlinear part; f is the type of nonlinearity that is used (sigmoid); x and y the system input and output respectively.

The derivation of the learning rules follows the anti-gradient method based on the regular energy (cost) function. The direct connections (A) and the nonlinear components (Q, R) are trained simultaneously.

The learning rules show that in the training phase the learning components (linear and nonlinear) are clearly separated, thus they may run with different training speeds. Applying second order methods to derive learning rules led us to some kind of cooperation between the linear and the nonlinear part.

Since one of the interested and still not formalized area of possible applications is the financial forecasting, the defined architecture has been used to forecast financial timeseries. Numerical applications validate the assumption: in stable markets (NYSE indexes) no significant contribution of the nonlinear term was found, indicating that the nonlinearities that are eventually present cannot be detected by this method. In the second case, with different data taken from an evolving market (BUX indexes), the nonlinear part gave substantial contribution to the network output.

On Extended Simple Eco-grammar Systems

Judit Csima

The concept of the eco-grammar system (the EG system, for short) has been introduced in [3] as a model of communities of agents which interact with their common shared environment. Several aspects of these systems were discussed in [4] and [7], properties of a restricted variant, called simple eco-grammar systems, were studied in [5], [1], [9], [2], [6], and [8]. Briefly, a simple eco-grammar system consists of several agents (represented by sets of context-free rules) and an environment (given by a set of $0L$ rules). At any moment of time, the behaviour of the system is described by the state of the environment which is a string over the alphabet of the system. The environmental state changes by derivation steps. In a derivation step the agents act on the string by applying one of their context-free rules - each agent rewrites only one letter - and the environment replaces, according to its $0L$ rule set, in a parallel manner the symbols where the agents do not perform any action.

Starting from an initial string representing the environment, a lot of sequences of strings following each other arise which describe the evolving system. The language generated by the eco-grammar system is the set of strings which can be obtained from the initial environmental state by a sequence of derivation steps. In the case of extended simple eco-grammar systems only those strings belong to the determined language which are over a distinguished subset of the alphabet of the system, the terminal alphabet. This notion was introduced and some basic properties were examined in [6].

Following that line, in this contribution we deal with a more sophisticated version of the derivation, the team behaviour of the extended simple eco-grammar systems: in each derivation step from the n agents exactly k or at most k have to perform an action. We describe the behaviour and the generative power of these systems according to some size parameters: the total number of agents, the number of agents being active in a derivation step. We examine the hierarchy of the language classes generated by extended simple eco-grammar systems with and without λ-rules.

The results demonstrate that while in the non-extended case the size parameters of the teams and the agent population have influence on the power of the system, in the extended case these parameters are not important: we obtain a collapsing hierarchy.

References

[1] J. Csima. Formal language theoretic models of ecosystems (In Hungarian). Master's thesis, ELTE, Budapest, 1997.

[2] J. Csima and E. Csuhaj-Varjú. Eco-rewriting systems: components with insertions and replacements. Manuscript, 1998.

[3] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, and Gh. Păun. Eco(-grammar) systems. A preview. In R. Trappl, editor, *Proc 12th European Meeting on Cybernetics and System Research*, pages 941–948. World Sci. Publ., Singapore, 1994.

[4] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, and Gh. Păun. A grammatical approach to likelife interactions. *Artificial Life*, 3:1–28, 1997.

[5] E. Csuhaj-Varjú and A. Kelemenová. Team behaviour in simple eco-grammar systems. Theoretical Computer Science, to appear.

[6] J. Dassow and V. Mihalache. Eco-grammar systems, matrix grammars and E0L systems. In Gh. Păun, editor, *Artificial Life: grammatical models*, pages 210–226. Black Sea Univ. Press, Bucharest, 1995.

[7] Gh. Păun, editor. *Artificial Life: grammatical models*. Black Sea Univ. Press, Bucharest, 1995.

[8] M. H. ter Beek. Teams in grammar systems. Master's thesis, Leiden University, Leiden, 1996.

[9] M. H. ter Beek. Teams in grammar systems: Hybridity and weak rewriting. *Acta Cybernetica*, 12(4):427–444, 1996.

The Components for Intelligent Tutoring Systems

Vladan Devedzic, Danijela Radovic and Ljubomir Jerinic

The concept of software components has received relatively little attention so far by the researchers in the field of AIED. This paper is an attempt to bring more light on this important concept and to describe the benefits that component-based ITSs can bring to the field. Surprisingly enough, there is still no consensus on the notion of component among AIED researchers. There are many open questions and unclear issues. By carefully structuring the description of software components in general, as well as by showing some examples of components for ITS design, the paper presents several important related issues, like functionality of components, their granularity, generality, interoperability, and reusability. Special attention is given to the architectural and communication considerations, as well as to the relation between components and ontologies for ITS design.

The purpose of this paper is threefold:

1. It is supposed to describe from different viewpoints (architectural, design, software engineering, and utility) the concept of software components that may be useful for development of Intelligent Tutoring Systems (ITSs).

2. It is also intended to be a survey of important problems, questions and issues related to such components.

3. It should draw the reader's attention to the possibilities that component-software technology can offer to the field of ITSs.

Although the concept of components has been largely used in the area of software engineering during the last decade (see, for example, [Adler, 1995]), it is only since recently that it draws significant attention in the community of researchers working in the area of Artificial Intelligence in Education (AIED). Exceptions from this rule include relatively few people from the AIED community (some examples are the work of Murray [Murray, 1996], Munroe [Munroe et al., 1994] and Devedzic et al., 1997]).

The paper has been inspired by an informal discussion on the topic of software components, held during the AIED'97 conference in Kobe, Japan . Starting from some conclusions of that discussion, as well as from the relevant literature both in the area of ITSs and from the area of software engineering, and from our own previous work, we have elaborated on the idea of software components for ITSs and organized the paper as follows. First the usual process of ITS development is overviewed, with some comments on its inherent difficulties. Then some requirements are enumerated that would significantly alleviate the process of ITS development. This is followed by a description and a discussion of the concept of software component and the most important issues related to it. Examples of software components that we have identified in our own work are also shown. Architectural and communication aspects of components for ITSs are shown next. The Discussion Section comments on the relationship between the concept of components and component of ontologies for ITS design. Finally, some conclusions and open questions are presented.

References

[1] Adler, R.M.: "Emerging Standards for Component Software", IEEE Computer, March 1995, pp. 68-76..

[2] Devedzic, V., Jerinic, Lj.: "Knowledge Representation for Intelligent Tutoring Systems: The GET-BITS Model", in: du Boulay, B., Mizoguchi, R. (eds.): "Artificial Intelligence in Education", IOS Press, Amsterdam / OHM Ohmsha, Tokyo, 1997, pp. 63-70.

[3] ErgoTech Systems White Paper, "Plug and Play Component Software for Manufacturing Systems", 1997, http://www.ergotech.com/ergotech/mofa.html.

[4] Munroe, A., Pizzini, Q., Towne, D., Wogulis, J., Coller, L.: "Authoring Procedural Training by Direct Manipulation", USC working paper WP94-3, 1994.

[5] Murray, T.: "Toward a conceptual vocabulary for intelligent tutoring systems", Working paper available at http://www.cs.umass.edu/ tmurray/papers.html, 1996.

Simulation Approach for Localizing Roots of Real Coefficients Complex Equation

Sasa Dimitrijevic, Bratislav Dankovic, and Dragan Antic

This paper presents a new approach for solving real coefficients complex equation based on simulation. The procedure is used for n-th order complex equation, in the form:

$$f(z) = a_n z^n + a_{n-1} z^{n-1} + \Lambda + a_1 z^1 + a_0 \tag{1}$$

where $a_i, i = 1, \Lambda, n$ are real coefficients, n is equation order and $z = x + iy$ is a complex variable. The technique relates solving complex equation:

$$f(x + iy) = 0 \tag{2}$$

what can be written in the form:

$$Re\{f(x + iy) = 0\} + iIm\{f(x + iy) = 0\} = 0 \tag{3}$$

The condition (3) is fulfilled if both absolute values of real and imaginary part are equal to zero:

$$\mid Re\{f(x + iy)\} \mid + \mid Im\{f(x + iy)\} \mid = 0 \tag{4}$$

Considering that this approach is used for localization of equation (1) roots, equation (4) model will be:

$$\varepsilon = \min_{x,y} \mid Re\{f(x + iy)\} \mid + \mid Im\{f(x + iy)\} \mid \tag{5}$$

The x, y values which correspond to minimum will be the roots of equation (1).

Simulation is conducted using block diagram simulation languages (SIMULINK, etc.) where $x = time$ and as a result of simulation value ε for $y = y_0 = const$ is obtained.

$$\varepsilon = \min_{y} \min_{x} \mid Re\{f(x + iy)\} \mid + \mid Im\{f(x + iy)\} \mid \tag{6}$$

Repeated simulations for different values of $y = y_0 = const$ are performed using MATLAB programming. When the desired accuracy is accomplished, the obtained values $x = x_0, y = y_0$, will represent one of the equation (1) roots. The procedure is then repeated for different values $y = y_0$. In that manner, all other roots can be localized.

References

[1] Karayanakis N. M., Computer-Assisted Simulation of Dynamic Systems with Block Diagram Languages, CRC Press, Boca Raton, 1993.

[2] Murray-Smith D. J., Continuous System Simulation, Chapman & Hall, London, 1995.

[3] Kuo F. F. ed., Computer Simulation of Dynamic Systems, Prentice-Hall, New Jersey, 1972.

Fuzzy Spark Advance Scheduling For Internal Combustion Engine

Sasa Dimitrijevic, Dragan Antic, and Predrag Stankovic

The authors have considered engine timing model which describes a simulation of a four-cylinder spark ignition internal combustion engine [1]. A new approach for dynamic control of spark advance using fuzzy logic based scheduler has been proposed. By introducing variable spark advance value, the set-point remains in the area of maximum torque developed by the engine.

The key elements of the engine model are: throttle, intake manifold, mass flow rate, intake to power stroke delay, torque function and engine speed. An internal combustion engine from the throttle to the crankshaft output can be modelled using *Matlab/Simulink*. Time based model is converted to a crank angle synchronized model by using a *Hit Crossing Block* which captures accurately discrete events such as the beginning of the intake stroke for each cylinder.

Fuzzy logic based controller has been introduced for the purpose of spark advance scheduling. Fuzzy controller is fired only once every 180 degrees using discrete blocks with sample time set to 1. Based on engine speed and mass of fuel-air mixture cylinder for combustion as fuzzy scheduler inputs, input normalization for their crisp values is performed. Using rule-term base, fuzzy logic rules and fuzzy inference procedure, spark advance fuzzy value can be determined. Fuzzy output value is then denormalized using centre of gravity method.

Considering that engine gain is variable (dynamical properties of the engine are not stationary) parameters of the PID controller in the control loop must be modified. The gain is dynamically normalized with the knowledge of spark advance value.

References

[1] SIMULINK Automotive examples, The MathWorks, Inc., 1996.

[2] Matlab - User's guide, The MathWorks, Inc., November, 1994.

[3] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, part I, II", IEEE, Trans. Syst. Man Cybern., Vol. 20, pp. 404-435, 1990.

[4] R. Babuska "Designing fuzzy control systems with MATLAB and SIMULINK", EUFIT 1993 - First European Congress on Fuzzy and Intelligent Technologies, Aachen, 1993.

Universal Characterisation of Non-Transitive Preferences

József Dombi and Nándor J. Vincze

Some decision support systems use preference relation and their final result is usually non transitive (for example the results of the French school: ELECTRE, PROMETHEE) Transitivity can be obtained by using the incomparability measure together with a cutting procedure.

The question of transitivity and non-transitivity appears on preference based decision structures. Using the classical utility approach the transitivity is always valid.

In multicriteria decision making there are two main research area. On one hand the preference approach will achieve the transitivity, because it gives a consistency and on the other hand the utility will establish intransitivity, because the human decision making has this property.

In classical utilities we order only one numerical value to an alternative and this is certainly transitive. But in the real life, there are many non-transitive structures, so the utilities need extension.

This was a long process (which has some crucial points, for example: Alois paradox, and the answers how the models can keep the linearity) in which the results was a new kind of utility concept. P. Fishburn introduced the skew-symmetric bilinear functional. In this model the non-transitive decision has an exact functional representation.

Our main result:

Introduction of the concept of the k-cyclically of preferences was considered in the structures of two-valued lotteries: to win $m with probability p, and $0 with probability $1-p$, denoted this lottery by $[m, p(m)]$. The k-cyclicity in this structure means (the preference relation is denoted by >):

$$
\begin{aligned}
&[m,p(m)] > [m+1,p(m+1)] > ... > [m+k,p(m+k)] > [m,p(m)]\\
&[m,p(m)] > [m+1,p(m+1)] > ... > [m+2k,p(m+2k)] > [m,p(m)]\\
&.\\
&.\\
&.\\
&[m,p(m)] > [m+1,p(m+1)] > ... > [m+lk,p(m+lk)] > [m,p(m)]
\end{aligned}
$$

In this structure we take the preference representation of P. Fishburn with skew-symmetric bilinear functional, and we suppose, that this functional can describe with real-valued function: $\varphi(x,y) = h(x-y), x \geq y$.

We can summarise our main result in the following theorem:

Theorem: For every $k, n \in N$, where $2k \leq n$ and for every $\varepsilon > 0$, there exists $j \in N$, such, that the preference > is k-cyclic on the interval $[j,n]$ and there exist k-cyclic preference function $h(m)$, for every k in the form $h(m) = v(m)g(m)$ where $g(m)$ is the positive solution of: $F(m+i)g(m) - g(m+i)F(m) = g(i)$ $1 \leq m, i, m+i \leq n$ functional equation, on the set $1, 2, ..., n$, when $F(m) = \frac{f(m)}{v(m)}$, for positive solution $v(m)$ of the functional inequality system:

$v(i) \leq v(m)v(m+i)$ if $i \neq rk$, and
$v(rk) > v(m)v(m+rk)$
where $r \in N, r \leq [\frac{f(n)}{2k}]$, and $j \leq m, m+i, rk \leq n$.

Machine Learning Methods in Synthesis of Four Bar Linkages

Anikó Ekárt

Kinematic synthesis of four bar mechanisms is an engineering problem that is difficult to solve by generative methods. The present approach is a variant based method, that makes use of two artificial intelligence paradigms: decision trees and genetic programming.

The goal of the paper is to obtain a partitioning of the mechanism-space, to give a structural description for the class of mechanisms likely to produce the desired coupler curve. The selection of the best suited mechanism can be done afterwards by browsing the class and analyzing its members.

Two methods for classification of four bar mechanisms are shown: decision trees generated by a variant of the C4.5 program by Quinlan [4] and genetic programming (introduced by Koza [3] as an extension to genetic algorithms).

From kinematic point of view a four bar mechanism could be designed for [5]: (1) path generation - the path of a tracer point on the coupler link is of interest, (2) rigid body guidance - the entire motion (path and angle) of the coupler link is of interest, and (3) function generation - the relative motion of links is of interest.

By analytical methods a four bar mechanism could be synthesized for at most five prescribed positions of the coupler point [5].

In the present paper the path generation problem is considered, but the results could be extended to the motion generation and function generation problem, too. The path generation problem consists in finding the appropriate structural parameters of the mechanism (length of links and position of coupler point) which generates a given path (coupler curve). The functional description consists of curve fragments that should be on the coupler curve. A curve fragment is given as a list of points, where the neighboring points are connected with straight lines. For each curve fragment a tolerance is specified.

Bose et al. [1] tries to solve the same problem by using case-based reasoning. He makes use of a multi-level case retrieval procedure followed by case adaptation. For the adaptation of cases he uses a set of very simple adaptation rules, that make slight modifications to one or two parameters of the mechanism. However, the approach is limited to particular classes of mechanisms.

This paper gives two methods for describing a class of four bar linkages generating similar coupler curves. In this case the adaptation step becomes simply a search for the best mechanism within a small class.

Starting from the desired curve fragments, the class of mechanisms generating coupler curves that might be close to these fragments is automatically created. First, a measure of similarity for two curves is defined, then a catalog of four bar mechanisms is browsed and the suited mechanisms are selected . Using an adapted version of the C4.5 program, a decision tree is generated for the chosen class. This decision tree gives in fact the permitted intervals for each structural parameter of any mechanism in the class.

The genetic programming based method generates a set of conditions that have to be satisfied by the members of the class. The conditions are mathematical relations among the structural parameters of the linkage, such as:

$$CouplerLink > FollowerLink + 0.5$$

The two methods are discussed in detail and a comparison with the other approaches is made.

References

[1] Bose, A., Gini, M. and Riley, D., A case-based approach to planar linkage design. Artificial Intelligence in Engineering 11 (1997) 107-119.

[2] Hrones, J. A. and Nelson, G. L., Analysis of the four bar linkage. Wiley, 1951.

[3] Koza, J. R., Genetic programming: On the programming of computers by means of natural selection. The MIT Press, 1992.

[4] Quinlan, J. R., C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

[5] Sandor, G. N. and Erdman, A. G., Advanced mechanism design: Analysis and synthesis, vol .2, Prentice Hall, 1984.

Analysing the Noise Sensitivity of Skeletonization Process

Attila Fazekas and András Hajdu

The necessity of designing skeletonization algorithms dates back to the early years of computer technology, to the 1950s. It was realised that in some applications (the first problem was the character recognition), it is enough to take only a reduced amount of information into account instead of the whole image, which is usually a line-drawing. The basic idea was to "peel" the original image by iteratively removing suitable contour points. This procedure is the so called skeletonization, which obtains a line-like shape (the skeleton), so the further analysis become more easily executable.

Many skeletonization algorithms have been analysed from several points of view in the past ten years to compare the results they produce. However it is very difficult to measure quantitatively the "goodness" of such a method. The analytical comparison of the methods is very sophisticated, since they are based on different models. This is the reason why the skeletonizations are compared according to the results they produce in the practice. There are papers about the technical parameters of these algorithms (like computation speed, memory requirement, etc.).

Our purpose is to analyse the "goodness" of skeletonizations from a special point of view. We found that the skeletonizations were not examined statisticaly (with a large number of experiments) to test how the noise corruption affects the extraction of the skeleton, and how the skeletonization processes can cope with noisy images. Unfortunately the input images are rarely ideal, but are corrupted with some kind of noise. For example, the contours in the image of a printed circuit board is often corrupted by a contour noise, which makes the contours disconnected, or thicker, etc.

Our purpose was to decide which is the most efficient algorithm for noisy images, among the investigated five one. A large database of 21000 skeletons was used to obtain performance indices for the algorithms. Linear correlation was detected between the level of the noise and the distance of the reference and test skeletons. The algorithms could be grouped according to their tolerance with respect to different types of noises and images. The calculated rank values of the algorithms may help someone to choose an algorithm which produces the most reliable result on a given type of image which is corrupted with a given type of noise. It seems to be interesting to go on with analysing other skeletonizations, which are based on other models, or to make investigations in higher dimension (3D).

The n-Distinguishable-Queens Problem, an Extension of a Classical AI Problem

Endre Fülöp

The n-queens problem is to place n indistinguishable objects (queens) on an n*n grid (chess board) so that no two objects are placed on the same row, the same column, or the same diagonal. Since this classical combinatorial problem has a simple and regular structure, it has been traditionally used as a test-bed to develop and benchmark new AI search problem solving strategies. In this paper we present an extension of the classical n-queens problem. We provide the queens with different personality, i.e. individual wishes to be placed on particular parts of the chess board. (For instance Queen No. n may prefer the black squares against the white ones.) So the solutions of the original problem will not have an uniform goodness. One implements the queenly desires better than the other. The new problem is to find the best of all solutions of the original n-queens problem, the solution, which satisfies the wishes as far as possible. To solve this extended constraint satisfaction problem we present an algorithm based on effective probabilistic local search strategies published in recent years by R. Sosic and J. Gu and we also describe a practical, engineering application where this model and algorithm can be used successfully (scheduling of university lectures and rooms).

A Fast Constant-Space Substring Search Algorithm

Harri Hakonen and Timo Raita

A family of fast constant-space substring search algorithms is described. The central idea behind the new scheme is a generalization of the well-known Boyer–Moore string searching approach complemented with a technique called q-slicing, a form of probabilistic q-gram matching. The search procedure is independent of the alphabet size and results in efficient and practical on-line implementations. Experiments show that the new variants are comparable to the fastest known Boyer–Moore methods.

Introduction. The substring searching problem is to find all occurrences of a pattern $pat[1..m]$ in a given text $text[1..n]$. The strings are concatenations of symbols taken from the input alphabet Σ of size σ. String searching has been studied extensively (see e.g. [1] for an excellent survey). As a result, several efficient and elegant solutions to this problem have been given. The most practical implementations have been derived from the seminal ideas of Boyer and Moore [4].

The original BM algorithm (BM for short) aligns the pattern with a text position, compares the corresponding symbols of pat and $text$ starting from the last symbol of the pattern and advancing to the left. If a mismatch (if any) is found, the pattern is shifted forward with respect to the text and the process is repeated. Prior to the actual search, two tables are formed in $O(m)$ time. These tables determine the length of the shift when a (mis)match is found. The *match heuristic* table determines how much the pattern must be moved in order to align an identical part of the pattern with the matched part of the text. The *occurrence heuristic* table expresses the position of the rightmost occurrence of each symbol of Σ in the pattern. Thus, the heuristic determines how much we can shift the pattern in order to align the mismatched text symbol with an identical pattern symbol.

The actual search procedure of BM consists of three distinct phases which are repeated until the text is exhausted: (i) fast skipping over non-matching text regions, (ii) match checking when some evidence of a pattern occurrence has been found and (iii) shift to the next position. All these steps have been the subject of refinements [5, 6, 8, 11, 13, 14]. A detailed analysis of various BM substep combinations can be found in [8]. In what follows, we will mainly concentrate on steps (i) and (iii), by giving an intuitive analysis on the length of the shift on the basis of the information used for those steps.

The well-known and widely used variant of Horspool [6] (BMH) discards the match heuristic due to its small practical significance with non-periodic patterns. This results in $O(mn)$ worst case complexity. However, on the average it is only $O(n/m)$ and therefore very fast. It is clear that on the average, BMH makes shorter shifts than the original BM due to the omission of the match heuristic. This behaviour is emphasized when σ is small. On the other hand, the role of the match heuristic becomes insignificant when σ becomes large (this is studied in more detail in [12]). As suggested in [11], we should try to compensate the omission of the match heuristic in other ways during the search. One alternative is to extend the occurrence heuristic for bigrams, incorporating thus both heuristics (at least partly) into one. This idea was introduced in [14] and was shown to give improved running times, especially for small input alphabets. Moreover, if we follow the idea of BMH and choose always (independently of the position where the mismatch occurred) the bigram composed of the text symbols aligning with $pat[m-1]$ and $pat[m]$, we obtain a very close approximation to the match heuristic. In the special case where the mismatch occurs at $pat[m-2]$, they are identical. Thus, if we can generalize the approach (as suggested in [2, 3, 10]) and use q-grams ($q > 2$) of arbitrary length, we obtain a heuristic which is a combination of both original ones. However, the disadvantage of the approach is that the preprocessing time and the space demand both increase rapidly, being proportional to σ^q.

The q-slicing method. In order avoid the excessive time and space requirements during preprocessing and still get large shifts, we combine two ideas. First, the information, on the basis of which the shift is made, is gathered from a large text region. For this reason, we call the symbol group thus obtained, a *generalized q-gram*. A q-gram is normally defined as a substring of q consecutive symbols; we relax this definition and require that it is a subsequence of length q. The symbols of the generalized q-gram are picked from the text during the search process. The positions of the symbols are defined in a *template* at the start of the preprocessing stage. What is significant in the definition of the template is, that we do not require the symbols to reside in the text 'under' the pattern.

Second, we reduce the size of the alphabet at the cost of losing some accuracy in symbol comparison. The idea is to partition the symbols of the input alphabet into equivalence classes and tag each class with a symbol of a reduced alphabet Σ'. Now, instead of comparing individual symbols, we compare tags of the classes. Currently, our implementations form the tag by taking the k least significant bits from the original symbol encoding; hence the name *q-slice*.

The search procedure starts by preprocessing the mapped pattern symbols as follows: choose a template T of length q. Gather all q-slices of pat defined by T and store each of them into a convenient machine-dependent unit (i.e. parameters k and q are chosen so that the q-slice fits into a byte or a word, for example). For each q-slice, compute a shift value, that is, the number of positions we can move the pattern with respect to the text without losing any matches. After the preprocessing, the search proceeds by mapping the text symbols to the reduced alphabet on-the-fly and using template T. Because the tag representation is short, we can efficiently use information which is scattered into a wide region in the neighbourhood of the current context. This gives a basis to increase the average length of a shift using only a small amount of comparisons. A drawback of the approach is, that at each q-slice match we must confirm that an identical symbol pattern has been found. In this respect, the tag sequence can be regarded as a special Rabin–Karp type signature [9]: the equality of two slices is a necessary, but not sufficient condition for the equality of the corresponding patterns. However, theoretical analysis shows that false matches occur rarely.

Summary. A new family of fast substring searching algorithms is devised by combining the concept of a q-gram with the relaxation of symbol equivalence and introducing templates. The strategy is called q-slicing and effectively, it makes sampling of the text on-line to skip fast over regions where the pattern $pat[1..m]$ cannot occur. In spite of the fact that most machine architectures do not support the idea of q-slicing on the hardware level, the efficiency of the new methods is comparable to the fastest known substring searching algorithms. The algorithms in this family are highly parametric and can thus be adapted according to special needs. Tests have shown that this approach results in a constant space algorithm which has an average shift length even larger than m, the size of the pattern itself.

References

[1] **Baeza-Yates, R.A.:** Algorithms for String Searching: A Survey, *SIGIR Forum, Spring/Summer 1989, Vol. 23, No. 3,4, pp. 34-58*

[2] **Baeza-Yates, R.A.:** Improved String Searching, *Softw. Pract. Exp., Vol. 19, No. 3, March 1989, pp. 257-271*

[3] **Baeza-Yates, R., Krogh, F.T., Ziegler, B., Sibbald, P.R. & Sunday, D.M.:** Notes on a Very Fast Substring Search Algorithm, *Comm. ACM, Vol. 35, No. 4, April 1992, pp. 132-137*

[4] **Boyer, R.S. & Moore, J.S.:** A Fast String Searching Algorithm, *Comm. ACM, Vol. 20, No. 10, October 1977, pp. 762-772*

[5] **Galil, Z.:** On Improving the Worst Case Running Time of the Boyer-Moore String Matching Algorithm, *Comm. of the ACM, Vol. 22, No. 9, September 1979, pp. 505-508*

[6] **Horspool, R.N.:** Practical Fast Searching in Strings, *Softw. Pract. Exp., Vol. 10, 1980, pp. 501-506*

[7] **Galil, Z. & Seiferas, J.:** Time-Space-Optimal String Matching, *J. Comput. System Sci., Vol. 26, 1983, pp. 280-294*

[8] **Hume, A. & Sunday, D.M.:** Fast String Searching, *Softw. Pract. Exp., Vol. 21, No. 11, November 1991, pp. 1221-1248*

[9] **Karp, R.M. & Rabin, M.O.:** Efficient Randomized Pattern-matching Algorithms, *IBM J. Res. Develop., Vol. 31, No. 2, March 1987, pp. 249-260*

[10] **Knuth, D.E., Morris, J.H. & Pratt, V.R.:** Fast Pattern Matching in Strings, *Siam J. Comput., Vol. 6, No. 2, June 1977, pp. 323-350*

[11] **Raita, T.:** Tuning the Boyer-Moore-Horspool String Searching Algorithm, *Softw. Pract. Exp., Vol. 22, No. 10, October, 1992, pp. 879-884*

[12] **Tarvainen, H.:** A Theoretical Framework for the Substring Searching Algorithms, *M.Sc. Thesis (in Finnish), University of Turku, Finland, May 1995*

[13] **Sunday, D.M.:** A Very Fast Substring Search Algorithm, *Comm. of the ACM, Vol. 33, No. 8, August 1990, pp. 132-142*

[14] **Zhu, R.F. & Takaoka, T.:** On Improving the Average Case of the Boyer-Moore String Matching Algorithm, *J. of Inf. Proc., Vol. 10, No. 3, 1987, pp. 173-177*

Investigations of Tubular Structures

Cs. Halmai, E. Sorantin, A. Tanacs, P. Winkler, and G. Wolf

Today's digital imaging modalities in Radiology enable computer postprocessing. Infrarenal aortic aneurysms, meaning the abnormal extension of the main abdominal blood vessel, represents a potential life threating medical condition. Since various theropatic options are available today, reliable prognostic parameters have to be obtained.

The aim of this paper is to demonstrate how advanced image processing can be applied to Spiral-CT data of patients with infrarenal aortic aneurysms in order to provide a basis for further medical management.

Algorithm: In the method presented in this paper at first the tubular structure of the aorta is extracted semiautomatically into an image sequence. Then a thinning algorithm is performed to get the skeleton. The user can decide the main path of the skeleton by using a 3D virtual reality environment. This path is smoothed and equidistant slices that are orthogonal to the path are extracted from the original object. The crossectional area of each slices are calculated and maximal diameters are estimated. These results are exported into diagrams for further treatment planning.

Software: In this project in-house 3D thinning algorithms were used. 3D pathes are stored in a 3D chain code format. A VRML editor utility CosmoWolrds and self made conversion tools converting from chain code to VRML and vice versa were used. Possible pathes in the skeleton were found by the help of graph algorithms. Path was smoothed with vector smoothing algorithm. Orthogonal slices were extracted and measurements were performed within the IDL (Interactive Data Language from Creaso Research Systems, USA) environment.

The results showed that the maximal diameter and crossectional area describes well the state of the blood vessel.

Training of Artificial Neural Networks by Linear Mappings

Miklós Hoffmann and Emőd Kovács

The different types of artificial neural networks are widely used in computer science. One of the special fields of applications is the computer graphics and geometry, where an interesting problem is the analytical description and handling of linear mappings. The purpose of this presentation is to evaluate the learning ability of the back-propagation method in linear mapping recognition. For better understanding let us consider a well-known affine transformation of the plane given by the equations

$$\tilde{x} = a_{11}x + a_{12}y + t_1 \quad \tilde{y} = a_{21}x + a_{22}y + t_2$$

where (x, y) and $(\tilde{x}, \tilde{y})$ are corresponding points. If we consider a classical two-layered perceptron with two input neurons and two output neurons, the weights and the thresholds can be given by

$$w_{ij} = a_{ij}, \quad \theta_i = t_i \quad (i, j = 1, 2)$$

where w_{ij} is the weight to the connection between the input node i and the output node j, while θ_i is the threshold connected to the output node i. This neural network produces the same transformation described above. How many pairs of points do we need to train this neural network starting with random weights? What can we say in terms of other kind of transformations, e.g. projective mappings, degenerated mapping of the space to the plane? These were the main questions of our project, where the latest results of the analytical description of central axonometric mappings were also applied.

The authors wish to thank the Hungarian National Scientific Fund (OTKA No. F019395) for their financial support of this research project.

References

[1] J.A.Freeman, D.M.Skapura: Neural Networks; Algorithms, Applications and Programming Techniques, Addison-Wesley, 1991.

[2] J.Hoschek, D.Lasser: Fundamentals of Computer Aided Geometric Design, AK Peters, Wellesey, 1993.

[3] J.Szabó, H.Stachel, H.Vogel: Ein Satz ber die Zentralaxonometrie, Sitzungsber., Abt. II, sterr. Akad. Wiss.,Math.- Naturw. Kl. 203, 3-11 1994.

On a Merging Reduction of the Process Network Synthesis Problem

Csaba Holló, Zoltán Blázsik, Csanád Imreh, and Zoltán Kovács

In a manufacturing system, materials of different properties are consumed through various mechanical, physical and chemical transformation to yield desired products. Devices in which these transformations are carried out are called *operating units*. Thus, a manufacturing system can be considered as a network of operating units which is called *process network*. A process design problem in general means to construct a manufacturing system. A design problem is defined from a structural point of view by the *raw materials*, the *desired products*, and the available operating units, which determine the structure of the problem as a *process graph* containing the corresponding interconnections among the operating units. Thus, the appropriate process networks can be described by some subgraphs called *feasible solutions* of the process graph belonging to the design problem under consideration. Naturally, the cost minimization of a process network is indeed essential where the *cost of a process network* is the sum of the costs of the operating units included in the process network considered. It is known that this problem is NP-complete, and therefore, each reduction of this problem has great importance.

In this talk, a new type of reduction is introduced which is based on the merging of operating units. The mergeable operating units are determined by an equivalence relation on the set of the operating units, and all of the operating units included in a class of the partition belonging to this equivalence relation are merged into one new operating unit. It is proved that this reduction procedure called *merging reduction* has the following property: an optimal solution of the original problem can be derived from an optimal solution of the reduced problem and conversely. The theoretical investigations providing this connection are equipped with an empirical analysis of the merging reduction on randomly generated problems which shows the measure of the size decrease under this reduction type.

Decomposition of CFT(S) Transformations with Look-ahead

Tamás Hornung and Sándor Vágvölgyi

We generalize Engelfriet's decomposition result: $T^R = T \circ LH$, where T^R, T, and LH denote the class of tree transformations induced by top-down tree transducers with regular look-ahead, top-down tree transducers, and linear homomorphisms, respectively (see [1]).

A top-down tree transducer can be considered as a nondeterministic recursive 'program' that acts on trees and generates trees. We can describe the 'program' as a grammar which operates on a special storage type. We consider Engelfriet's theorem replacing top-down tree transducers by CFT(S) transducers, that is by context-free tree transducers which operate on a storage type S.

In a regular tree (RT) grammar (see [5]), the nonterminals have rank 0. The context-free tree (CFT) grammars are obtained from RT grammars by allowing nonterminals of rank greater than 0. The CFT grammars can be considered with two modes of derivation: in side-out (call by name) and outside-in (call by value). A CFT grammar with these derivations is said to be inside-out tree (IO) grammar and outside-in tree (OI) grammar (cf. [3]).

The concept of storage type is introduced in [2] and [4]. Roughly speaking, a storage type S consists of a set of input elements and a set of configurations. The input elements can be encoded as configurations. The configurations can be tested by predicates and transformed by instructions.

Let MOD be the set RT, IO, OI, CF of modifiers where CF abbreviates the type of context-free grammars. Let X range over MOD. An X(S) transducer is an X grammar in which every rule is provided with a predicate, and every nonterminal of the right-hand side of the rule has an instruction. Considering a derivation of the X(S) transducer, each occurrence of a nonterminal A is associated with a configuration c (different occurrences may be associated with different configurations). A rule of the X(S) transducer can be applied to the tuple A(c) as a rule of an X grammar can be applied to the nonterminal A, provided the test specified by the rule holds for c. The new configurations for the nonterminals of the right-hand side of the rule are obtained by transforming c according to instructions also specified in the rule. The initial nonterminal of the grammar is associated with a configuration corresponding to an input element. Thus, the X(S) transducer induces a transformation from the input set to the set of terminal trees or strings. The class of transformations induced by X(S) transducers is also denoted by X(S) (cf. [4]).

Top-down tree transducers are RT(TR) transducers, where TR is a particular storage type, called the tree storage type. For this storage type, the root of trees can be tested and the trees can be transformed into their immediate subtrees. Top-down tree transducers are the same as RT(TR) transducers.

The concept of storage type S with look-ahead is introduced in [2] and [4] as a generalization of regular look-ahead, and denoted by S_{LA}. The storage type S_{LA} is obtained from S by adding special tests to the set of predicates of S, so-called look ahead tests. They have the form <B>, where B is a CF(S) transducer. The look-ahead test <B> is true on a configuration c if and only if the transducer B can derive a terminal string from $B_{in}(c)$ where B_{in} is the initial nonterminal of B. We allow Y(S) transducer s in the look-ahead tests (where Y is in MOD) and we write SY instead of SLA in this case (SLA = SCF). Since the class of domains of top-down tree transducers is equal to the class of regular tree languages (see [5]), Engelfriet's decomposition result takes the form RT(TRRT) = RT(TR) o LH. (1) In this paper, we study whether the equation X(SY) = X(S) o LH (2) is true for a storage type S and for the modifiers X, Y in MOD when X differs from CF. We show that X(S) o LH is a subclass of X(SX), but there exists a storage type S such that X(S) o LH is not equal to X(SX). An X(SY) transducer, in which look-ahead tests do not occur in the negated part of tests belonging to the rules, is denoted by X+(SY). The class of transformations induced by X+(SY) is denoted also by X+(SY). It holds for all storage types S and modifier X in MOD CF that X+(SX) = X(S) o LH. (3) Since RT+(TRRT) = RT(TRRT), (3) implies (1), and if X+(SX) = X(SY) then (2) holds true. We show that X(SIO) = X(SRT) = X(SLA) for all storage types S. We give examples for storage types such that X+(SX) = X(SX).

References

[1] J. Engelfriet, Top-down tree transducers with regular look ahead, Math. Systems Theory 10 (1977) 289-303.

[2] J. Engelfriet, Context-free grammars with storage, Technical Report Nr 86-11 (1986), University of Leiden, July 1986.

[3] J. Engelfriet and E.M. Schmidt, IO and OI, I., II., J. Comput. System Sci. 15(1977) 328-353, 16(1978) 67-99.

[4] J. Engelfriet and H. Vogler, Pushdown machines for the macro tree transducers, Theoret. Comput. Sci. 42(1986) 251-368.

[5] F. Gécseg and M. Steinby, Tree automata (Akadémiai Kiadó, Budapest, 1984).

Construction of Recursive Algorithms for Polarity Matrices Calculation in Polynomial Logical Function Representation

Dragan Janković

Compact representation of switching functions is not only the matter of notation convenience, but highly relates to the analysis and synthesis of these functions. Both analysis and synthesis procedures, as well as final realizations, can be greatly simplified by choosing appropriate representations of switching functions.

In the case of Reed-Muller (RM) expressions, the problem to determine the most compact representation reads as determination of optimal polarity for switching variables. By choosing between the positive or negative literals for each variable, but not both at the same time, the Fixed polarity RM (FPRM) expressions are defined [5].

In a FPRM, the number of products, or equivalently, the number of non-zero coefficients may be considerably reduced by choosing different polarities for the variables. The FPRM with the minimum number of products is taken as the optimal FPRM for f. If there are two FPRMs with the same number of products, the one with the smaller number of literals in the products is taken.

There is no method to determine apriori the polarities of variables for a given function f. In practice, it is necessary to generate all the FPRMs and chose the optimal one. That can be efficiently done by generating the polarity matrices $\mathbf{P}_{\mathrm{RM}}$ whose rows are RM-coefficients for the given f with different polarities of variables. The efficiency of generation of $\mathbf{P}_{\mathrm{RM}}$ is based upon its recursive structure originating in the Kronecker product representation of the RM-transform matrix.

Polynomial representations of Multiple-valued (MV) functions are very interesting with advent of multiple-valued circuit technology, in particular recent experience with current-mode circuits that are very attractive for implementation of MV functions. Specially, the realization of 4-valued corresponding circuit is very efficient. The problem of compact representations is even harder in the case of MV functions. Galois field (GF) expressions are generalization of RM-expressions to MV case [7]. Optimization of GF-expressions can be studied and solved in a way similar to that used for RM expressions. In particular,efficient methods for generation of polarity matrices $\mathbf{P}_{\mathrm{GF}}$ for GF-expressions of ternary functions are reported in [6], while the correspond methods for quaternary functions are reported in [3], and further elaborated in [1], [2], [4].

Reed-Muller-Fourier expressions are an alternative extension of RM expressions to MV case [8]. It has been shown that RMF expressions require on the average smaller number of products than GF expressions to represent a given function f[9]. The optimization of RMF expressions is performed in the same way as in the GF-expressions by choosing different polarities for the variables. As in the case of RM and GF expressions, there are no methods to determine apriori the polarity for the variables in a given f to get the RMF expression with the minimum number of products. For that reason, the efficient calculation of polarity matrices is a very important task. An analyse of present recursive methods for calculation of polarity matrix for some particular expressions shows that recursive approaches are more efficient than others methods. Therefore, the construction of recursive relations for polarity matrix calculation for various expressions is a very interesting problem.

In this paper, we uniformly consider the coefficients in various expressions for logic functions as spectral coefficients in particular spectral transforms. We show that polarity matrix can be generated as convolution of f with columns of related transform matrix. The recursive properties of the polarity matrix result from properties of the convolution matrix. We give a unique method to construct recursive procedures for the polarity matrices calculation for any Kronecker product based expression of MV functions.

This method involves existing methods as particular cases and permits various generalizations. For illustration, we derive two recursive algorithms for calculation of fixed polarity Reed-Muller-Fourier expressions for vour-valued functions.

References

[1] Falkowski, B.J., Rahardja, S., "Efficient algorithm for the generation of fixed polarity quaternary Reed-Muller expansions", *Proc. 25-th Int. Symp. on Multiple-Valued Logic,* Bloomington, 1995, 158-163.

[2] Falkowski, B.J., Rahardja, S., "Efficient computation of quaternary fixed polarity Reed-Muller expansions", *IEE Proc.-Comput.Digit.Tech.,* Vol.142, No. 5, 1995, 345-352.

[3] Falkowski, B.J., Rahardja, S., "Fast construction of polarity coefficient matrices for fixed polarityquaternary Reed-Muller expansions", *Proc. 5th International Workshop on Spectral Techniques,* Beijing, China, March 1994, 220-225.

[4] Falkowski, B.J., Rahardja, S., "Quasi-arithmetic expansions for quaternary functions", *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design,* Chiba, Japan, 1995, 265-272.

[5] Green, D.H., "Reed-Muller expansions with fixed and mixed polarities over GF(4)", *IEE Proc.-Comput.Digit.Tech.,* Vol.137, No. 5, September 1990, 380-388.

[6] Harking, B., Moraga, C., "Efficient derivation of Reed-Muller expansions in multiple-valued logic System", *Proc. 22nd IEEE Int. Symp on Multiple-Valued Logic,* Sendai, Japan, 1992, 436-441.

[7] Muzio, J.C., Wesselkamper, T.C., *Multiple-valued Switching Theory,* Adam Hilger, Bristol, 1986.

[8] Stanković, R.S., Moraga, C., "Reed-Muller-Fourier representations of multiple-valued functions over Galois fields of prime cardinality" *Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design,* Hamb urg, Germany, Sept. 1993, 115-124.

[9] Stanković, R.S., Janković, D., Moraga, C., "Reed-Muller-Fourier versus Galois Field representations of Four-Valued Logic Functions", *Proc. 3rd Workshop on Applications of the Reed-Muller Expansion in Circuit Design,* September 19 -20, 1997. Oxford, UK, 269-278.

Object Oriented Model for Parameter Estimation of Partially Separable Functions

Jaakko Järvi

Introduction. In parameter estimation the goal is to fit a model function that depends on adjustable parameters to a set of observed data. This is done by defining a merit function (e.g. mean square error) to measure the agreement between the data and the model function and by minimising this function. This article focuses on *structural* or *partially separable* model functions, where the actual model function is a sum of *component* functions, e.g. a spectrum consisting of a sum of spectral lines.

Numerous algorithms has been presented for model fitting tasks in the literature but usually from the numerical analysis viewpoint, treating the model as a plain vector of parameters plus a function. However, the natural representation of the model may be highly structural consisting of several component functions. Moreover the component functions may have common parameters or other dependencies between their parameters. The flat model representation is therefore inconvenient for the user and it is the application developers task to provide a conversion to and from the structural representation.

In this article an object oriented model is presented to serve as an intermediate link between the above two representations, providing simultaneously the interface for the optimisation algorithms and the structured view for the user.

Object oriented model. The model is a collection of classes comprising a core to represent structured model functions. These core classes implement the basic structural and flat views to the model function, as well as the mechanisms for function value and derivative calculations. The core also handles the possible relations between the parameters. The extension of the core model for a specific application is done by providing a class for each type of component function. Only member functions for calculating the values of the component functions are required in these classes. The model utilises *automatic differentiation* [1] to compute the derivatives, so explicit code for analytical derivatives is not needed. By exploiting the partial separability of the model functions, automatic differentiation can be implemented efficiently.

The core classes implement all the functionality needed for manipulating component functions and their parameters. The user interface for this task can therefore be built solely based on the core classes. Adding new classes to the model hierarchy does not cause any need for changes in the interfacing code. An example of a user interface built in this manner is given.

In the structural representation of model functions inheritance and dynamic binding is used. Therefore we loose something in computational efficiency. Efficiency of the model is discussed and it is shown, that the speed decrease compared to low level code is not very significant. The persistence, i.e. the ability store and retrieve the objects of the model is also considered.

The model was developed while working on nuclear magnetic resonance (NMR) spectra estimation. A case study of NMR-spectral fitting is therefore conveyed through the article to clarify the ideas presented.

The are scantly descriptions of object orientation together with parameter estimation in the literature. Related work can be found from [3, 4] containing descriptions of computer systems sharing some similarities with our model. See [2] for the description of the NMR analysis software built using the object oriented model presented here.

Conclusions. Traditionally object oriented programming is not very popular in numerical codes, where efficiency is of utmost importance. However, by careful design the penalty of replacing low level codes with more abstract ones is quite moderate. The main result of this paper is to show that using object oriented principles a flexible and extendible framework for expressing structural functions can be built without sacrificing efficiency in the evaluation of the function values and derivatives.

References

[1] Rall L.B.: Automatic differentiation: Techniques and Applications, Lecture Notes in Computer Science 120, Springer-Verlag, Berlin 1981

[2] Järvi J.: A PC-program for automated analysis of NMR spectrum series, Computer Methods and Programs in Biomedicine **52** (1997) 213-222

[3] Majoras R. E., Richardson W. M., Seymour R. S.: An object-oriented approach to evaluating multiple spectral models, Journal of Radioanalytical and Nuclear Chemistry **193** (1995) 207-210

[4] van Tongeren B. P. O., Boxman R. D. C., Deumens J. W., van Leeuwen J. P., Mehlkopf A. F., van Ormondt D., de Beer R., QUANSIS, An object-oriented data-analysis system for in vivo NMR signals, Journal of Magnetic Resonance Analysis, **2** (1996) 75-84.

Difference Functions of Dependence Spaces

Jouni Järvinen

Z. Pawlak introduced his notion of information systems (sometimes called knowledge representation systems) in the early 1980's. Information concerning properties of objects is the basic knowledge included in information systems, and it is given in terms of attributes and values of attributes. For example, we may express statements concerning the color of objects if the information system includes the attribute "color" and a set of values of this attribute consisting of "green", "yellow", etc. In general, an information system is determined by specifying a set U of objects, a set A of attributes, and an indexed set $\{V_a\}_{a \in A}$ of value sets of attributes. Each attribute is a map $a \colon U \to V_a$, which assigns to every object a value of the attribute a.

In an information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ each subset of attributes B $(\subseteq A)$ defines an indiscernibility relation $I(B)$. This relation is an equivalence on the object set U such that its equivalence classes consist of objects which have the same values for all attributes in B. An attribute set C $(\subseteq A)$ is a reduct of B, if C is a minimal subset of B which defines the same indiscernibility relation as B. Hence, the reducts of an attribute set are its minimal subsets defining the same partition of objects. The reduction problem means that we want to enumerate all reducts of a given subset.

Reduction problem can be studied in a more simpler algebraic structure called dependence space. Here we characterize the reducts by the means of dense families of dependence spaces. Dense families are important since they contain enough information about the structure of dependence spaces. Dependence spaces induced by indiscernibility relations are also studied. We show how we can determine dense families of dependence spaces induced by indiscernibility relations by applying indiscernibility matrices.

We present an algorithm for finding the reducts of any given subset of a dependence space. The algorithm is based on the notion of difference function which connects the reduction problem to the general problem of identifying the set of all minimal Boolean vectors satisfying an isotone Boolean function.

The Cost-Effective and Component Based Intelligent Tutoring Shell - the GET-BITS Model

Ljubomir Jerinic and Vladan Devedzic

Many important issues in design and implementation of ITSs (Intelligent Tutoring Systems) some functionality focuses on the way inter-object connections are represented, manipulated, and stored in the computer. However, the advancement of AI methods and techniques makes understanding of ITSs more difficult, so that the teachers are less and less prepared to accept these systems. As a result, the gap between the researchers in the field of ITSs and the educational community is constantly widening. While ITSs, also called knowledge based tutors, are becoming more common and proving to be increasingly effective, each one must still be built from scratch at a significant cost. Also the present ITSs need quite big development environments, huge computing resources and, in consequence, are expensive and hardly portable to personal computers. We have been searching for efficient ways to do these knowledge-engineering tasks. This paper describes our efforts toward developing uniform data and control structures that can be used by a wide circle of authors, e.g., domain experts, teachers, curriculum developers, etc., who are involved in the building of ITSs.

Applying Fuzzy Multiple Criteria Optimization to PCB Scheduling

Tommi Johtela, Jouni Smed, Mika Johnsson, and Olli Nevalainen

In this paper we describe an actual production environment for *printed circuit board* (PCB) assembly. Our focus is on one of the various work phases of the production line, namely *surface mounting*. In our previous work we have applied heuristic and exact methods for solving the problem. However, it is hard for these methods to account for the existence of several conflicting criteria present in the problem. Therefore, in the present study we apply fuzzy techniques. This approach enables us to analyze the production environment and its properties more closely and to build a more complete but still manageable model of the production process. In addition, the fuzzified criteria correspond more accurately to the requirements of the user.

A typical assembly line for automatic component printing on PCBs comprises several successive work phases. Each PCB goes through a glue dispenser, two surface mounting machines, an oven and a manual insertion phase. The number of jobs processed on the line is very high and the amount of PCBs in a job is usually quite small. Therefore, the *set-up times* form a significant part of the total production time. The meeting of the due-dates is in this case managed by a two-level priority classification (i.e., products are either urgent or non-urgent). The overall production time is affected by the fact that there are two different widths for the PCBs, and the change of the conveyor width causes an interrupt in the printing process. Also, some PCBs require component printing on both sides, and in order to avoid unnecessary storaging, the other side should be printed as soon as possible after the first side. The components are either glued or pasted to the circuit board, and therefore they may require different oven temperatures.

Because the sum of different component types on a PCB is significantly smaller than the capacity of the feeders in the machine, we can choose an appropriate input organization quite freely. We have chosen a set-up which is identical for only a part of the products, and therefore the products to be manufactured must be divided into groups.

In our earlier work we have developed several methods (e.g., *heuristic algorithms*) for solving the grouping, but our solutions lacked a measure which takes into consideration the various aspects of the actual production environment. By using a “classical” objective function we were able to find a grouping with a minimal number of groups and after that affect somewhat the distribution between the groups. Urgencies, conveyor widths, the management of the double sided PCBs, oven temperature and the size of the set-up were all ignored. Although the original heuristics improved the actual production radically, further refinements are still needed—and one easy way to include them to the model is to apply fuzzy techniques. The fuzzy approach allows us to build a model which considers all the criteria and, at the same time, remains understandable. As a benefit of this the user has now a much clearer notion of the software model and its operation.

All the criteria characterizing a good solution can be taken into account by representing each criterion as a fuzzy set. The intuitive idea behind this is: the greater the membership of the solution in the set, the better the solution. The objective function is obtained from the aggregation of the fuzzy sets representing different criteria. Thus, the objective function includes every criterion affecting the solution. It is also possible to specify conflicting goals where different criteria draw the solution to different directions. The final solution is essentially a *compromise* among all the criteria. Also, the priorities among the criteria have to be considered. The priorization can be done by *weighting* the fuzzy sets. The weights ensure that the more important criteria have a greater effect on the value of the objective function than the less important ones.

The primary objective is clearly *minimizing the number of groups* since the set-up times are the bottleneck of the production. Our first approach was to model also this goal as a fuzzy criterion. The problem, as it turned out, was that the relative importance of the criterion had to be set so high that its weight dominated all the other weights. That in turn narrowed the effective range of the other criteria and their contribution to the solution diminished.

We overcame this problem by using a solution computed by our earlier algorithm as a basis and then improving the solution with a heuristic guided by the fuzzy criteria. We could thus fix the number of groups, which made our task much easier. Also, the distribution of weights became more even and the effect of the less important criteria became notable when evaluating our solutions.

The advantage of the fuzzy model is that it is easy to develop and to update (e.g., to add a new criterion). It also provides us with a simple basis for the design of a graphical user interface: the user can decide the importance of the criteria and therefore influence the forming of the groups. The implementation of a flexible and easy-to-use GUI is critical, and this kind of interactivity was hard to accomplish in the original approach. Also, the user has now a better and more intuitive grasp of the solutions provided by the fuzzy system.

The Structure of the Univoque Set

Gábor Kallós

The specification of the univoque numbers is one of the newest fields in the research of generalized number systems. In [1] and [2] Z. Daróczy and I. Kátai have specified the univoque sequences and have presented a method for the computation of the Hausdorff dimension of the univoque set in the cases $1 < \beta \leq 2$, where β is the base of the number system. Now we continue this investigation in the general case, where $\beta > 1$ is the base number of an arbitrary number system, $\Theta = \frac{1}{\beta}$. The first part of this investigation is based on the methods presented in [1] and [2] but later significantly new approaches are needed. In the sequel we work only on the set of the fractions.

For $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots) \in \{0, 1, \dots [\beta]\}^N$ let $\langle \varepsilon, \Theta \rangle = \sum_{n=1}^{\infty} \varepsilon_n \Theta^n$. A sequence ε is said to be univoque (with respect to be univoque (with respect to Θ), if $\langle \varepsilon, \Theta \rangle = \langle \delta, \Theta \rangle$ is only true if $\varepsilon = \delta$, i.e. $\varepsilon_n = \delta_n$, for $n \in N(\delta \in \{0, 1, \dots, [\beta]\}^N)$. In this case the number $\langle \varepsilon, \Theta \rangle$ is said to be univoque, too. The quality $x = (\varepsilon(x), \Theta)$ is called the regular expansion of x, where $\varepsilon(x) = (\varepsilon_1(x), \varepsilon_2(x), \dots)$ and $\varepsilon_i(x)$ is maximal. We denote the set of univoque sequences and the set of regular sequences by $U(\Theta)$ and $R(\Theta)$, respectively. It is true in general, that the set of univoque sequences is symmetrical and self-similar. We can specify the univoque sequences by using the statement that a sequence $\varepsilon \in \{0, 1, \dots, [\beta]\}^N$ is univoque with respect to $\Theta \iff \varepsilon, \underline{[\beta]} - \varepsilon \in R(\Theta)$, where $\underline{[\beta]} - \varepsilon$ is the complementary sequence of ε. Thus, in order to decide whether a sequence and that of the complementary sequence. To establish that an expansion producing a number less than 1 is regular, we use the result of W. Parry [3].

After this we break down the problem into two cases. If the fraction part of the base number is smaller than a suitable bound, then the structure of the univoque sequences is relatively easy to describe. In the other case the situation is much more complicated, but eventually we can present a method for the computation of the Hausdroff dimension of the univoque set in this case, too. We illustrate the theoretical results with interesting examples.

References

[1] Z. Daróczy, I. Kátai, Univoque Sequences, *Publ. Maht. Debrecen,* 42. (1993), 397-407.

[2] Z. Daróczy, I. Kátai, On the Structure of Univoque Numbers, *Publ. Math. Debrecen,* 46. (1995), 385-408.

[3] W. Parry, On the β Expansions of Real Numbers, *Acta Math. Hung.,* 11. (1960), 401-406.

Module Based Reinforcement Learning for a Real Robot

Zsolt Kalmár, Csaba Szepesvári, and András Lőrincz

The behavior of reinforcement learning (RL) algorithms is best understood in completely observable, finite state- and action-space, discrete-time controlled Markov chains. In contrary, robot-learning domains are inherently infinite both in time and space, and are partially observable. Here we suggest to break up the problem into overlapping subtasks and design controllers for each of the subtasks. Then, operating conditions are attached to the controllers (together the controllers and their operating conditions are called modules), and possible additional features are designed to facilitate observability. A new discrete time-counter is introduced at the "module-level" that clicks only when a change in the value of one of the features is observed. The interactions of the controllers are analyzed (several tools are proposed for doing this), and RL is proposed to learn an optimal switching strategy for switching between the modules. The approach was tried out on a real-life robot. Several RL algorithms were compared by ANOVA and it was found that the model-based approach worked significantly better than the model-free approach. The learned switching strategy performed equally well as a handcrafted version. Moreover, the learned strategy seemed to exploit certain properties of the environment which could not have been seen in advance, supporting the view that adaptive algorithms are advantageous to non-adaptive ones in complex environments.

Questions on complex dynamic security

Zoltán Kincses

Security is the art of conscious risk taking. The meaning of this definition is that it has to be invested more in a system's security than the cost of evading this security. It is obvious that there is no 100security, even if the security of the system is increased up to a non-usable level of the system. In case a secure and easy to operate system is needed, the possibility of auditing is an essential feature, in order to detect the invader during or leastwise after the attack. The right tool for solving this problem is the sure identification.

Sure identification means identifying the individuals and not only the terminals or passwords. This can be ensured through **biometry**.

The secure (authenticated) communication between extremities must be alive during the whole life-cycle of the transmitted information. This can be resolved by **cryptology**. We have different algorithms proved mathematically that security depends on the key- handling. This key must be secure stored, portable, easy to use, and not detected by any recently known way from the stored format.

Due to different platforms the main problem is the platform dependency. The spreading of the Java programming language makes possible to prepare applications based on the idea of "write once, run anywhere!"

The requirements can be resolved through a tool, called **smart card**. Based on ISO 7816 standard it is possible to use modern smart cards that are able to generate 1024 bit length RSA keys, store biometrics data for sure identification, etc. Their security is well-organized against both software and hardware attacks. Due to Java Card specification it is possible to create **platform independent** applications.

In the paper I intend to show some problems and results in comparison of the ideas described above by creating the *complex dynamic security of a smart card based system*.

There are several components that have some contact points to other components, but the integrity of security of the whole system must be ensured during the mixing operation. On the other hand complex security must handle flexibly every arising changes.

When analyzing different security systems it can be seen the lack of a general framework description in planning - developing - applying processes. Such a framework description should take also into consideration some exceptions, like identification of people with different deficiencies. Embedding exceptions handling should not decrease the security of the system.

As well the proved functioning a complex security system must have also a representation for better illustration. Beside the above mentioned this will also be presented in the full paper.

On the Optimal Parameters of a Sinusoidal Representation of Signals

András Kocsor, László Tóth, and Imre Bálint

One of the most useful parametric models in the spectral resolution of digital signals is the modelling by a sum of phase-shifted sinusoids in form of $\sum_{n=0}^{N-1} A_n \sin(\omega_n t + \varphi_n)$, where $A_n,\ \omega_n$ and φ_n are the component's amplitude, frequency and phase, respectively. This model generally fits well speech and most musical signals due to the intrinsic nature of the representation functions. However using all of the above parameters leads to a very difficult optimization problem. The solutions are generally based on eigenvalue decomposition, but this is computationally very expensive and works only if the sinusoids and the residual signal is statistically uncorrelated. To find the best approximation in a subspace of the least dimension N is a major problem. To speed up the representation process some authors use rather ad hoc methods for determining the parameters. Such is the model of McAulay and Quatieri, which looks for peaks in the FFT spectrum. In the present paper the BFGS optimization method is applied to find the best approximating subspace of minimal dimension N, which is determined by parameters $\{A, \omega, \varphi\}$ and ensures a mean square error of approximation below a preset threshold.

From the invention of the telephone, speech or more generally voice processing especially voice representation has a paramount importance in electrical engineering. In the last years the rapid development of multimedia and computer networks brought a revival of the high-effective coding and representation problem. By the classic model of speech generation, the voiced part of the speech comes from the oscillation of the vocal chord, which can be modelled by an oscillating string. The voice is consisting of a fundamental and it's harmonics, which is well representable in the form $\sum_{n=0}^{N-1} A_n \sin(\omega_n t + \varphi_n)$ used in the paper. The error of approximation gives the 'unvoiced', noise-like part, which can be decoupled from the signal. The model fits well also musical signals, since the voices of most musical instruments (stringed-, wind instruments, etc.) consist of harmonic sinusoids. The rest-signal contains again the noise-like parts of the voice (as the drum), which should be modelled separately. The voice representation is used for data compressing, as well as detecting voiced parts, pitch estimation, or modification of the time-scale of the music, etc.. The above form of the model yields a complicated optimization problem enforcing some simplifications. In case of DFT, the number of sinusoids and their frequencies are fixed providing a rapid way of computing amplitudes and phases. However the individual components (sinusoids) of the representation generally differ from the really voice forming components. The method of [McAulay-Quatieri] tends to deduce the real frequencies of components from the results of DFT. A basically different approach characterizes the methods using eigenvalue decomposition. Here only the dimension N of the approximation subspace is fixed, but the algorithms are generally very slow and can be used only if the representation functions (sinusoids) and the rest-signals are statistically independent.

A detailed definition of the problem First the signal is sampled at points of a time-interval

$$\tau_0, \tau_1, \cdots, \tau_{K-1} \in [0, \tau]$$

and the obtained values are represented by the real sequence,

$$x[\tau_0], \cdots, x[\tau_{K-1}].$$

We are looking for the smallest number N, for which the function

$$\sum_{n=0}^{N-1} A_n \sin(\omega_n t + \varphi_n)$$

approximates the measured sample with the preset error $\epsilon > 0$,

$$\min_N \left[\min_{\substack{A_1,\cdots,A_{N-1}\\ \omega_1,\cdots,\omega_{N-1}\\ \varphi_1,\cdots,\varphi_{N-1}}} \sum_{k=0}^{K-1} \left(\sum_{n=0}^{N-1} A_n \sin(\tau_k \omega_n + \varphi_n) - x[\tau_k] \right)^2 < \epsilon \right]$$

The optimization problem with the Homogeneous Sinusoidal Representation Function A function will be given, which significantly simplifies the optimization problem defined in (1.1.1). Let be introduced the following notations,

$$w_k(\mathbf{A}, \omega, \varphi) := \sum_{n=0}^{N-1} A_n \sin(\omega_n \tau_k + \varphi_n), \quad k = 0, \cdots, K-1,$$

where

$$\mathbf{A} := [A_0, \cdots, A_{N-1}]^\top, \quad \omega := [\omega_0, \cdots, \omega_{N-1}]^\top, \quad \varphi := [\varphi_0, \cdots, \varphi_{N-1}]^\top.$$

Applying the identities,

$$w_k(\mathbf{A}, \omega, \varphi) := \sum_{n=0}^{N-1} A_n \sin(\tau_k \omega_n + \varphi_n) =$$

$$\sum_{n=0}^{N-1} A_n (\sin(\omega_n k) \cos(\varphi_n) + \cos(\omega_n k) \sin(\varphi_n)) = \sum_{n=0}^{N-1} a_n \sin(\omega_n k) + b_n \cos(\omega_n k),$$

where

$$a_n = A_n \cos(\varphi_n), \quad b_n = A_n \sin(\varphi_n).$$

Notice that,

$$\frac{b_n}{a_n} = \tan(\varphi_n), \quad \varphi_n = \arctan\left(\frac{b_n}{a_n}\right), \quad A_n = \frac{a_n}{\cos(\arctan\left(\frac{b_n}{a_n}\right))}.$$

Let be introduced,

$$\frac{c_n}{\sqrt{c_n^2 + d_n^2}} := \sin(\omega_n), \quad \frac{d_n}{\sqrt{c_n^2 + d_n^2}} := \cos(\omega_n).$$

By using these notations,

$$\sum_{n=0}^{N-1} a_n \sin(\omega_n k) + b_n \cos(\omega_n k) =$$

$$\sum_{n=0}^{N-1} a_n \sin(\tau_k \arcsin(\frac{c_n}{\sqrt{c_n^2 + d_n^2}})) + b_n \cos(\tau_k \arcsin(\frac{c_n}{\sqrt{c_n^2 + d_n^2}})) = w_k(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}),$$

where

$$\mathbf{c} = [c_0, \cdots, c_{N-1}]^\top, \quad \mathbf{d} = [d_0, \cdots, d_{N-1}]^\top.$$

After all these steps, let be used

$$\mathbf{x} := [x[\tau_0], \cdots, x[\tau_{K-1}]]^\top, \quad \mathbf{w} = [w_0, \cdots, w_{K-1}]^\top$$

and the function to be optimized is

$$L_{\mathbf{xw}} := \mathbf{x}^\top \mathbf{x} \mathbf{w}^\top \mathbf{w} - (\mathbf{x}^\top \mathbf{w})^2.$$

This function will be called Homogeneous Sinusoidal Representation Function (HSRF). The properties of HSRF will be investigated by optimizing several artificial and natural test-functions based on the BFGS algorithm.

Error Diagnosis in Prolog Programs, A Critical View

Gabriella Kókai

Programming environments which include program development, program analysis and program debugging tools ([2]) are essential for the acceptance of programming languages as Prolog or other logic programming languages as a "real" one. Especially testing which is a process of executing a program with the intend of finding error (see [10]), and debugging which goal is to analyse the program to locate and fix the detected error (see [6]) are central issues of programming environments.

Research on this topic started with the development of Shapiro's APD-System ([12]). The algorithmic program debugging method introduced by Shapiro can isolate an erroneous procedure if a program and an input on which it behaves incorrectly is given. Shapiro's model was originally applied to Prolog programs to diagnose the following three types of errors: termination with incorrect output, termination with missing output and nontermination. A major drawback of this debugging method is the great number of queries put to the user about the correctness of intermediate results of procedure calls.

Many other debugging techniques and tools of Prolog programs were developed. A few of them base on Shapiro's algorithm (APROPOS [9], EDS [5], DED [8] and ADAss [1]) or used other methods (PRESET [14], RD [11], OPIUM [2] and PTP [4]) but the main disadvantage of these systems is that they can handle only *pure* Prolog programs. It means that these systems cannot be applied for *realistic* programs. For example let us take the following short Prolog program which parses simple arithmetic expressions and computes the result:

```
expression(Value) --> term(A), "+", expression(B), {Value is A + B}.
expression(Value) --> term(A), "-", expression(B), {Value is A - B}.
expression(Value) --> term(Value).
term(Value) --> atomicexpression(Value).
term(Value) --> atomicexpression(A), "*", term(B), {Value is A * B}.
term(Value) --> atomicexpression(A), "/", term(B), {Value is A / B}.
atomicexpression(Value) --> number(Value, _).
atomicexpression(Value) --> "(", expression(Value), ")".
digit --> "1"; "2"; "3"; "4"; "5"; "6"; "7"; "8"; "9"; "0".
number(Value, Expo, C, D) :- 'C'(C,Code,E), digit(C,E),
      (Expo = 10 -> Value is Code - "0", D=E;
             Value is Code * B + A, Expo is B * 10, number(A, B, E, D)).
compute(Value, A, B) :- expression(Value, A, B).
```

The systems listed above cannot handle the extensions of Prolog used in this program. This paper describes a new system developed for algorithmic debugging and functional testing of Prolog programs based on Shapiro's work . This system is able to find the error in a program containing the following

- Declarations
- Control facilities: if (->), or (;), repeat
- Definit Clause Grammars
- Handling Built-In Predicates

The task of the system is the following:

- Develop a grammar for Prolog. With the help of this grammar parse the program and generate a tree for the syntax. From the tree produce a prooftree in the programming language C which produces the graphical representation for the graphical user interface (see Figure 1)
- The modifications of Shapiro's debugging methods that they can handle the whole special features of ISO Prolog

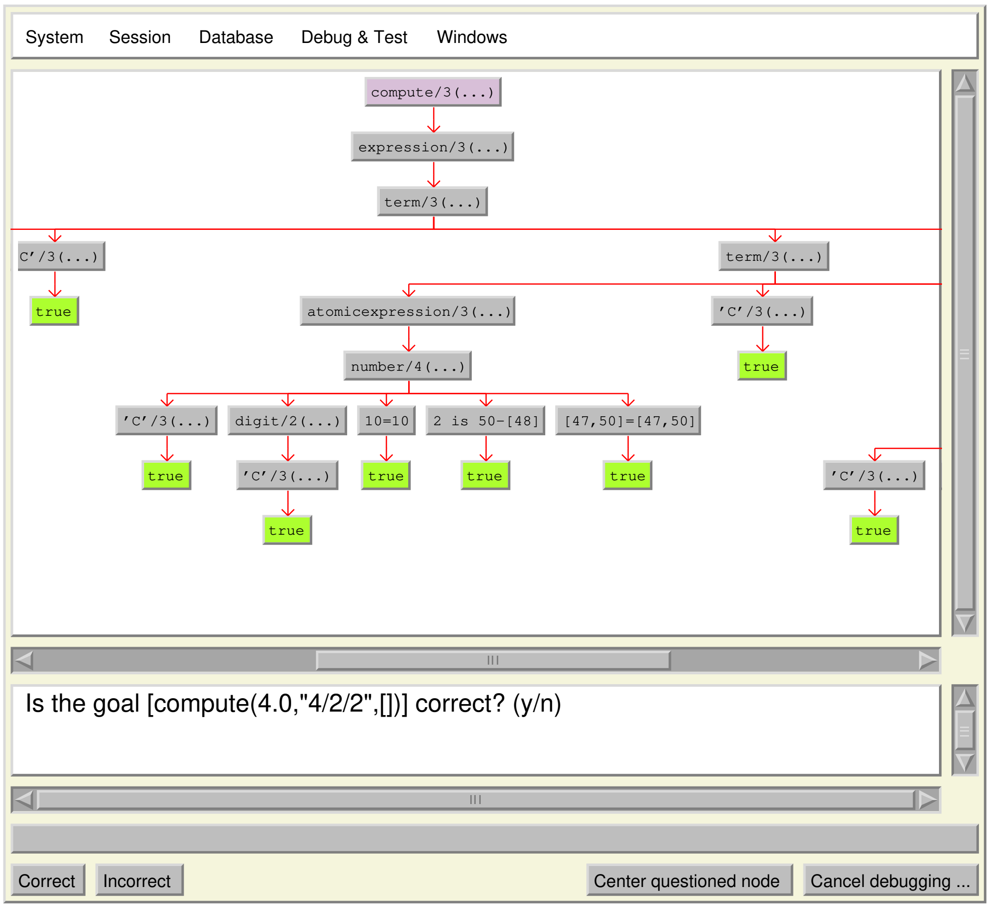


Figure 1: The generated prooftree in main window of the system

References

[1] Drabent, W., Nadjm-Tehrani, S., Małuszyński, J.: *Algorithmic Debugging with Assertions*, In Proc Meta-Programming in Logic Programming, 1988, 375-378,

[2] Ducasse, M.: *Opium - An Extensible Tracer for Prolog*, Technical Report Lp14, January 1987, 203-210

[3] Ducasse, M. Noye J.: *Logic Programming Environments Dynamic Program Analysis and Debugging*, Journal of Logic Programming, 19/20, 1994, 351-384

[4] Eisenstadt, Marc: *Retrospective Zooming: A Knowledge Based Tracing and Debugging Methodology for Logic Programming*, In Proc of the 9.th IJCAI, 1985

[5] Ferrand, Gérard:, *Error Diagnosis in Logic Programming an Adaptation of E.Y. Shapiros's Method*, Journal of Logic Programming, 1987, 177-198

[6] Kamkar M., Shahmehri N., Fritzson P.: *Bug localization by algorithmic debugging and program slicing*, In Proc. of the Programming Language Implementation and Logic Programming Symposium, Linköping, Sweden, 1993, Deransart and Małuszyński, LNCS 456 Springer

[7] Kókai, G., Harmath, L., Gyimóthy, T.: *Algorithmic Debugging and Testing of Prolog Programs* ICLP '97 The Fourteenth International Conference on Logic Programming, Eighth Workshop on Logic Programming Environments Leuven, Belgium, 8-12 July 1997, 14-21

[8] Lloyd J. W.: *Declarative Error Diagnosis*, New Generation Computing, 1987, 133-154, OHMSHA LTD and Springer Verlag

[9] Chee-Kit Looi: *Analysing Novices' Programs in Prolog Intelligent Teaching System*, In Proc of the European Conference on Artificial Intelligence, Munich 1988, 314-319,

[10] Myers. G. J.: *The art of software testing Business data processing* John Wiley and Sons, 1979

[11] Pereira, L. M. : *Rational Debugging in Logic Programming* In Proc of the Third Logic Programming Conference, London, England, July 1986, 203-210

[12] Shapiro, E. Y.: *Algorithmic Program Debugging* MIT Press (1983)

[13] Sterling, L., Shapiro, E., Y.: *The Art of Prolog* The MIT Press Cambridge Mass. 1986

[14] Takahashi, H., Shibayama, E.: *PRESET - A Debugging Environment for Prolog* , Proc of the International Conference on Logic Programming, Tokyo, 1985

Some Aspects of Financial Instruments' Price Modelling

Prof. Borko Krstic, Ognjen Radovic, Srdjan Marinkovic, and Ksenija Dencic

The necessity of mathematically describing some of the most complex and most unusual phenomena has started the searching for a "model" ai. for some mathematical rules that are capable of production some parts of the reality using computers. The classical attitude of complex system observation has been based on the universum reduction in the set of independent, separate entities, that are introverted and unable for real communication. Nonlinear dynamics is, on the other hand, concentrated to dynamics of the system on the whole. Nonlinear dynamic systems (NDS) are systems in a state of motion and systems whose behavior can not be expressed by linear algorithm. However, real world is a world consisted of endless heterogenities and complexities, one multidimension world that doesn't recognize straight lines and completely symmetrical forms; in this world, even empty space is squwed.

The scientists have pointed out to the fractal dimension of capital markets and brought back interest for nonlinear models. The idea of fractal nature of time series of many economic phenomena has been present for a long time, but it has been neglected because of the complex mathematics which one is in the connection with it. The application of complex nonlinear models and the interest for the fractal distribution has increased with the development of high technology. Fractal distribution is primarily the consequence of different sentiments of the investors as well as the result of time factor influence on decision making process.

Thanks to the fact that the scientist begin to understand better and better nonlinear systems, their complexity and selforganization, a new paradigm is emerging. We begin to understand complexity as natural state, not as deviation. Many physical systems are, in fact, complex nonlinear open systems. Simple, isolated systems (that are in the centre of attention of traditional science), present the extreme idealization.

APT (Ross & Ross, 1980.) has been wide used in portfolio management as alternative model for capital-asset pricing CAPM (Sharpe). A difficulty with APT arise when it should estimate expected stock's return. The fact that there is a set of factors with such characteristics that expected returns can be explained as linear combination of every asset's exposition to these factors, is the key idea of APT. Last years, artificial neural networks, instead of classical statistical techniques, has been used for prediction in APT model. The result that have been obtained using neural networks show better performance comparing to the linear regression for the predictions out-of-sample.

The aim of the authors has been to show some difficulties in the use of classical pricing models and to cite some modern tendencies in a nonlinear dynamics systems analyse (where as well belong financial instruments pricing). In the developed program (OLIMP 1.0), beside classic statistical approach to the stock price volatility estimation, the authors have used chaos theory, fractal analyse of signals and neural networks.

A New Look at Reverse Polish Notation

Predrag V. Krtolica

Reverse Polish notation was introduced by Polish mathematician Lukasievicz in 1958. By this notation, the arithmetic expression was noted in postfix manner, instead of using the usual infix notation. For example, the infix expression

$$a + b$$

in reverse Polish was noted like

$$ab + .$$

The algorithms for transforming postfix expression to the infix one, and vice versa are well known. However, these algorithms concern only the arithmetic expression with four fundamental binary arithmetic operators. The reason for this lies in the fact that the reverse Polish notation was aimed for computer architectures which had used stack for computing values of the expression. At the machine level all operations are fundamental. Therefore, reverse Polish notation could be useful for many other problems. If one wants to find derivative of the expression symbolically, he should make the tree of the expression. Then, he could make the derivative tree of the original expression tree. Making the expression tree begins with the transformation of the input expression in postfix notation. If we are limited to fundamental arithmetic operations, we are actually limited to polynomial or rational functions. In qualitative analysis of nonlinear differential equations (i.e. dynamic systems governed by these equations) there is a need for processing expression which include some forcing law (usually sine or cosine law). Software which accepts arbitrary differential equation should provide facilities for symbolic derivations of expression which contains binary, unary and function operators.

In this paper, the extension of the reverse Polish notation, as well as the extension of the corresponding algorithms for transforming infix expression to the postfix one and vice versa, are suggested. This extension allows processing the mathematical expressions containing not only fundamental arithmetic operators but also the unary operators and functions. It is shown that this could be extended on n-ary functions.

These improvements could be very useful in symbolic manipulations with expressions which contain not only the four fundamental arithmetic operators. A program using these improved algorithms is also developed.

Improved Greedy Algorithm to Look for Median Strings

Ferenc Kruzslicz

The distance of a string from a set of strings is defined by the .sum of distances to each string of the given set. A string that is closest to the set is called the median of the set. To find a median string is NP-Hard problem in general, so it is useful to develop fast algorithms that give a good approximation of the median string. These methods are significally depend on the type of distance used to measure the dissimilarity between strings. This algorithm is based on edit distance of strings, and constructing the approximate median in a letter by letter manner.

Introduction. If optical character recognition (OCR) problem is considered as a "black box" process, where images are mapped to strings, then we use a certain kind of off-line approach. In this way the efficiency of some OCR processes could be increased in OCR software and language independent manner. Suppose we have a set of strings as results of OCR processes of the same input bitmap. When the same OCR software was used to produce this set, with different paper orientation, changed resolution or simply repeated OCR processes we can eliminate the effects of pollution (fingerprints on the glass etc.) While in case of different OCR software their efficiency can be compared to each other.

To find a median string that is minimal in sum of distances form a given input set of strings, is known to be NP-hard problem. Therefore it is interesting to find fast algorithms, that give as good approximations. One of the latest algorithms is called greedy algorithm, because it builds up the approximate median string letter by letter, by always choosing the best possible continuation. In this paper an improvement of this algorithm is described.

The real advantage of the improved algorithm appears when the probability of edit operations in the garbling process is increased. In other words the improved algorithm works better if the strings in the input set are far from each other. In the example the string *recognition* was garbled with delete, insert and substitute string-edit operations. For substituting and inserting only the letters *r, e, c, g, t, i, o, n, s, p, a* were used, and each of the operations and its place was equally distributed.

For example let us consider the following test set H = { *ggroeonitin, rpcsogngapaoponc, secsgttin, gecciicn, eectgcgnitiopn, repsogniporpassn, raatnini, rnrcpnto, nirnscogtipntgo, nrectogansinageine* } where the cost of all edit operations is 1.

- The greedy algorithm gives the result *recrognitin* with summarized distances 75,
- and the improved algorithm found the string *rectognitin* with sum of distances 74,
- while the median of H is *recognition,* where the sum of distances is 73.

Conclusions. The improved approximate median algorithm is a simple refinement of the greedy algorithm. It has the same time complexity $O(k^2n|\Sigma|)$ as the previous one (where $|\Sigma|$ is the size of the alphabet, n is the number of input strings, and k is the length of the longest one). The space complexity was a bit reduced as well, because the new algorithm runs only in $O(kn)$ space. The garbled strings are closer to each other the improvement is less significant. Therefore the new improved greedy algorithm is more suitable for searching approximate median of highly dissimilar strings.

On the Partial Correctness of the Alternating Bit Protocol

Attila Lakatos and Pál Tőke

One of the basic problems of the computer communication is to give a reliable data transmission on an unreliable data transmission service. In the layered architectures like TCP/IP and ISO OSI Reference Model the data link control and transport layeres are to solve this problem. There are several protocol designs which can be applied. The common basic principle in these protocols is the retransmission and acknowledgement for the messages sent in individual protocol data units. >From mathematical point of view the simplest form of the data transmission phase can be modeled by the alternating bit protocol which can be described by the following parallel program over an appropriate $\mathcal{L}_P$ first order logical language.

$$
\begin{array}{rl}
\Psi \equiv \textbf{initial} & nextinput = u_0 \wedge nr = 1 \wedge \\
 & \wedge\, ls = mn = a = 0; \\
\textbf{cobegin} & \\
\textbf{loop} & \\
\alpha_0: & \\
 & \textbf{if}\ ls = a\ \textbf{then} \\
 & ls := ls \oplus 1; d := nextinput\ \textbf{fi}; \\
\alpha_1: & \\
 & send(ls, d)\ to\ (mn, inf) \\
\textbf{end} & \\
\| & \\
\textbf{loop} & \\
\beta_0: & \\
 & \textbf{if}\ mn = nr\ \textbf{then} \\
 & nextoutput := inf; \\
 & nr := nr \oplus 1\ \textbf{fi}; \\
\beta_1: & \\
 & send(nr \oplus 1)\ to\ (a) \\
\textbf{end} & \\
\textbf{coend} &
\end{array}
$$

The specification of the program is the following:

$$- \alpha_1 \to \circ[\quad (mn, inf) = (ls, d) \vee \\ (mn, inf) = (error, error)]$$

$- \alpha_1 \wedge P \to \circ P$
for every P-formula P
not containing *mn* and *inf*.

$- \beta_1 \to \circ(a = nr \oplus 1 \vee a = error)$

$- \beta_1 \wedge P \to \circ P$
for every P-formula P not
containing a-t.

$$- \alpha_0 \wedge ls = a \wedge \quad nextinput = u_i \wedge ls = ls_0 \to \\ \circ(nextinput = u_{i+1} \wedge \\ ls = ls_0 \oplus 1 \wedge d = u_i)$$

$- \alpha_0 \wedge P \to \circ P$
for every P-formula P not
containing *ls* and*d*.

$- \alpha_0 \wedge ls \neq a \wedge P \to \circ P$
for every P-formula P.

$$- \beta_0 \wedge \quad mn = nr \wedge nr = nr_0 \to \\ \circ(nextoutput = inf \wedge nr = nr_0 \oplus 1)$$

$- \beta_0 \wedge P \to \circ P$
for every P-formula P not
containing *nextoutput* and *nr*.

$- \beta_0 \wedge mn \neq nr \wedge P \to \circ P$
for every P-formula P.

The partial correctness of the protocol can be formulated by the following assertions:

$$\vdash_{\Sigma_{TP\Psi}} \quad start_\Psi \to \\ inf = u_0 \mathbf{atnext}(\beta_0 \wedge mn = nr) \tag{1}$$

$$\vdash_{\Sigma_{TP\Psi}} \quad \beta_0 \wedge mn = nr \wedge inf = u_i \to \\ inf = u_{i+1} \mathbf{atnext}(\beta_0 \wedge mn = nr) \tag{2}$$

The paper gives a detailed and new proof for these assertions and the possible generalization of the applied temporal logical methods for more complex cases are discussed.

Limiting Distortion of A Wavelet Image Codec

Joonas Lehtinen

An important feature of the *embedded zerotree wavelet* image coding algorithm (EZW) introduced by J. M. Shapiro is the progressivity of coding: the bits of the wavelet coefficients are sent in the order of importance in terms of *mean square error* (MSE). This enables the coder to stop when a predefined output size limit is reached. There are applications where certain level of image quality must be guaranteed. This can be achieved by overestimating the size of the output, but a more intuitive technique would be to *calculate the MSE while coding* and to output only the information that is needed to reach acceptable image quality.

In this paper a *new EZW based wavelet coding scheme is introduced*. Here we refer to the new coding scheme with name *Distortion Limited Wavelet Image Codec* (DLWIC). The codec is designed to be *simple to implement*, *fast* and have *modest memory requirements*. It is also shown, how the distortion of the result can be calculated while progressively coding a transformed image, if the transform is unitary.

EZW exploits spatial inter-band correlations of the wavelet coefficient magnitudes by coding the bit-planes of the coefficient matrix in a hierarchical order. The order is defined by a quad-tree structure where the completely zero subtrees (zerotrees) can be represented by only one symbol. In DLWIC the correlations between different orientations are also taken into account by binding together the coefficients on three different orientation bands in the same spatial locations. The maximum absolute values of the coefficients in all subtrees are stored in two-dimensional heap structure. This allows the coder to test the zerotree property of a subtree with only one comparison. A binary arithmetic coder with multiple separate probability distributions (states) is used to reach compression performance that is similar to the previously known EZW variants.

A biorthogonal wavelet transform is used to construct an *octave band composition*. Because biorthogonal wavelet transforms are unitary, the MSE of an image is equal in spatial and wavelet domains. The value of the square error (SE) is updated in the compression process. We start by calculating the initial SE as the total energy of the image and decrease it while bits are sent according the information content of the bits. For every bit sent, the change in SE of the image is defined by the difference of coefficients predicted values before and after the bit is sent. Calculations can be implemented efficiently with table lookups. This has the advantage that we know the MSE of the final decompressed image already in the coding phase and we can stop the transmission of bits when an acceptable distortion level is reached.

The efficiency of the DLWIC is compared to an advanced EZW variant and the industry standard JPEG using a set of test images. An estimation on speed and memory requirements of the DLWIC algorithm is made.

Structural Views in Object-Oriented Databases

Marijana Lomic

In the last period object-oriented databases are getting more and more publicity. They are expected to provide at least the functionality of relational database systems, because relational database management systems have been used for some time and users already got used to all the things that they provide.

Three level schema architecture consists of three schemata. These are: storage schema, conceptual schema and external schema. The research has given a lot of results towards bigger understanding of the conceptual and storage schemas for object-oriented database systems. Unfortunately external schema (in other word views) has not been studied as detailed as other schemata.

Views in the relational data model have proved themselves as a necessity in order to provide logical data independence. External schema can be changed by using views to meet individual needs of different users.

There are two kinds of view definitions: operational view definition and structural view definition. Operationally defined views are being defined as a result of a query. In this way we choose from the database a set of objects which satisfy a certain predicate. This is the kind of view definition, which is the base of the view definition system in relational database systems. There are some problems with it, especially with update of views and the propagation of the update into the base relations.

Structurally defined views are based on the model concepts of object-oriented data model. We define views by using inheritance (specialization and generalization). With help of these mechanisms we define the view to be derived from the base class as a subclass. In this way in the view we have the same objects (and eventually some new ones) but possibly with different attributes and behavior. Here derived class is interpreted as view class, only. Usually some derive operator is needed, which includes functionality of specialization and generalization. The functionality of generalization gives the possibility of information hiding, while specialization gives the possibility of adding some new information into the view.

In this paper the comparison is given between these two different kinds of view definitions and it will be shown that structurally defined views give the important contribution to semantic, functionality and usage of object-oriented database systems.

On the Exact Solution of the Euclidean Three-Matching Problem

Gábor Magyar, Mika Johnsson, and Olli Nevalainen

The *Euclidean 3-matching problem* (E3MP) is to form l disjoint triplets from $n = 3l$ points in the plane and connect the three points of each triplet by two line segments so that the total length of the line segments is minimal, see Figure 1.

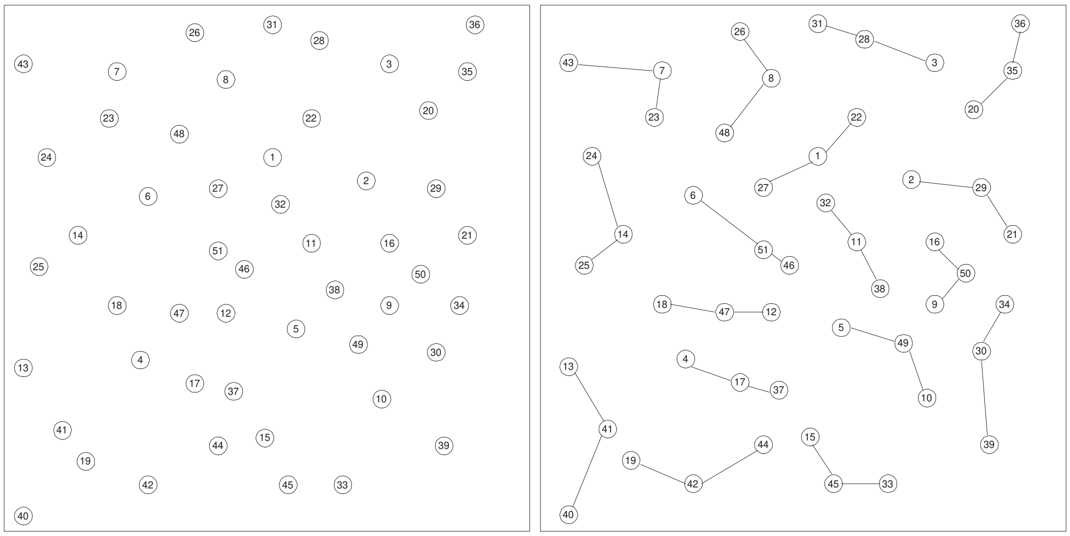


Figure 1. A sample 51-point 3MP problem instance ('eil51' from TSPLIB) along with its optimal solution. The vertices are numbered as they appear in the problem file.

The E3MP occurs in some industrial applications; in manual insertion of electronic components on a printed circuit board, the operations are arranged into close triplets in order to aid the worker in his task. Certain flexible machines (General Surface Mounter) for the automated insertion of electronic components have three to eight insertion heads and operate in cycles of component pickups and insertions with the heads. The throughput of the machine can be maximized by minimizing the length of the inter-board head movements. A similar problem occurs in the scheduling of an automated assay analysis instrument (AutoDelphia).

The problem is a special case of the general 3MP, which is NP-complete. The Euclidean problem can be solved heuristically by standard approaches, like local search heuristics using pairwise interchanges, simulated annealing, tabu search and genetic algorithm. Several lower bounds can be given for the problem. By knowing the optimal solution to the problem we can evaluate the quality of the lower bounds and the upper bound solutions of the heuristic approaches.

The 3MP can be formulated by the following *0-1 Integer Programming Problem*:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} c_{ij} \text{ subject to } \frac{1}{2} \sum_{i=1}^{n} x_{ij} + \sum_{k=1}^{n} x_{jk} = 1, (j = 1, \dots, n), x_{ij} \in \{0, 1\}$$

where the decision variable $x_{ij} = 1$ if and only if there is an $i \to j$ edge present in the solution, and c_{ij} is the cost of the $i \to j$ edge.

We find the optimal solution for the E3MP by the well-known *branch-and-bound* (B&B) technique. The branch-and-bound method implicitly enumerates all the feasible solutions of the particular combinatorial optimization problem under consideration. This is achieved by maintaining an enumeration tree which has partitions of the solution space in its nodes. The process starts with one node, namely the root, which represents all the feasible solutions to the problem. During this intelligent search, the *branching* *operation* is used to explore a selected node and to partition the solutions represented by it into its descendant nodes. The effectiveness of the B&B search method lies in the *bounding operation*, where we calculate lower bounds (in case of a minimization problem) for the solutions represented in the nodes. If the lower bound for the solutions of a certain node exceeds a known upper bound of the problem, the node and thus its whole subtree can be discarded as it cannot contain any better solution than already known. A good combination of a clever branching operation and a sharp lower bounding operation, augmented with some heuristic method for providing tight upper bounds can give reasonable performance for hard optimization problems. The task of writing an efficient branch-and-bound algorithm for our problem is challenging in itself and we discuss a number of design alternatives and their trade-offs in the study.

We introduce two branching strategies and three bounding operations for the E3MP. The latter include a lower bound based on the Lagrangian relaxation of the above integer programming formulation, a sharp problem-specific and a rather trivial lower bound. Both of the branching operations perform the partitioning by fixing and/or deleting certain edges (variables) in the *partial solutions* stored along with the particular node. These operations can use various edge selection rules for the branching and they can also incorporate automatic fixing and/or deleting of certain type of edges depending on the previous decisions leading to the partial solution in the particular node. We build several different B&B procedures from the above mentioned components and complete them with various node and edge selection rules and upper bound heuristics.

The role and performance of the different components of the B&B procedures is evaluated by empirical comparisons. The test set includes various problem instances, either coming directly from industry, or picked from TSPLIB, or generated randomly. The best variant is also compared to a public domain LP-solver and the results show that our procedure has a far better performance. We have also performed empirical tests on problems with restrictions on the available edges for the solution. These experiments provide useful information for a better understanding of the structure of the optimal solutions.

Summarizing our work, we have solved a restricted version of a practically relevant NP-complete graph problem (which is probably also NP-complete). The results show that our method is applicable for relatively small problems ($n < 54$) and its performance is superior to existing solving methods. The optimal solution of the problem is very useful at the evaluation of the lower bounds and the heuristic approaches to the problem. Furthermore, we suppose, that many of the ideas of our algorithms could be applied successfully to solve other graph problems.

Combination of Conformance, Performance and Interoperability Testing for Internet Applications

Mazen Malek and Roland Geche

Theoretically, conformance yields interworking and interoperability. This was the main driving force behind OSI systems. By having a glimpse at its methodology, conformance testing addresses the tested system with an exhaustive set of test cases, those of unwontedly complex structure and layout. Internet community, on contrast, was invented to overcome those drawbacks, and possibly eliminate the underlying complexity of conformity verification. This, unquestionably, eases real procurements and makes Internet so familiar. Because of such point of views, and due to the limited types of exchanged messages, conformance testing phase is not evaluated. Nevertheless, unconformed systems are sources of unaccepted functioning. Especially, when applications try to include sophisticated features, like persistent connection over TCP/IP stack or cache operations in proxy systems. This puts such applications on edge, and makes implication of their interoperability a hope rather than a culmination.

As a matter of fact, applicability of OSI conformance testing, generation of test suites, performance and interoperability testing of Internet protocols are fairly unfetched areas. The verification of interoperability is the process of demonstrating a successful communication between tested systems. This has a couple of views. First view is considering it as an absolute concept. Consequently, each system will have a certain measure to show its interoperability with similar ones. Second view is defining systems interoperability by performing pairwise tests. Performance testing, however, stands for testing system's capabilities in terms of traffic and load characteristics. In Internet community, the foregoing terminology stands for server's successful fulfillments of client requests. Verifying such requirements should be thought within the context of real time communications.

This paper, which tries to establish a framework for testing Internet implementations, will put previously untouched areas under focus. It starts with establishing a good study for testing interpretation on OSI-bases. This will contain, also, testing schemes and architectures. Afterward, the various interoperability requirements between different implementations will be shown. Additionally, an efficient step-by-step based testing algorithm is argued. This algorithm is cost-effective one and of great importance. The remainder of this paper use practical communication protocols, like HTTP, to describe illustratively these concepts

A Study of Portability in the Deployment of WWW

András Micsik

Several problems are experienced daily in the case of handling compound hypermedia WWW documents. One class of these problems, namely portability is studied in this paper in detail, and possible solutions are given using a new container architecture, the Portable HyperMedia (PHM) format. Structure and operations on PHM are discussed and connections with recent efforts in the field of metadata are explained.

Keywords : hypermedia documents, document-like objects (DLO), WWW, portability, Warwick Framework, Dublin Core, Internet

Introduction

As the World Wide Web [WWW] spread the world from the beginning of this decade, it incorporated more and more powerful tools and formats, and the information served via WWW became more and more complex. The content and layout of WWW pages become competitive with printed material, and in other aspects WWW pages have far more potential than printed documents. The meaning of document in case of the WWW is changing. WWW documents are sometimes more similar to a piece of software than to printed material. They may contain animations or has an annotation facility, and what is most important they are linked together.

Hypermedia documents has rich history. One can think about popular applications like ToolBook or HyperCard. However a large part of the functionality in these software are repeatable with current Internet technologies like WWW, CGI and Java. One significant difference between the two solutions is that while a document prepared with ToolBook can be easily moved to another computer, a similar WWW document may lose most of its functionality when moved to another location. One cause for this is that hypermedia documents on the Internet usually consist of hundreds of files. But very often WWW-based documents are still desired to use as single documents, to send, to store, to manipulate easily.

A new term appeared for such documents: Document-like Objects (DLO) [Ferber96]. DLOs has a complex nature:

- they may contain files in lots of different formats: text, graphics, animation, video, audio and 3D (VRML)
- they may contain executable parts (JAVA and CGI)
- their files and data are interconnected with links.

In this paper a way for evolution of Internet hypermedia documents is shown, using a new container format, the Portable Hypermedia (PHM) coupled with metadata description and automatic learning techniques. The PHM container does not obsolete current WWW usage and file formats, but enhances the manageability of hypermedia documents on the Web. The main advantage of this approach is the seamless integration of this new format into the current Internet without requesting the rewriting of the current tools.

Furthermore the PHM format is also capable for solving problems with the description and cataloging of available data on the Internet, pointed out by the Dublin Metadata Workshop [WGMD95].

In Section 2 common problems in current usage of WWW documents are examined, setting the requirements for current needs (Section 3), and proposing a new container format, the Portable Hypermedia (PHM) for the Internet society. The solutions offered for the problems listed in Section 2 are readable in Section 4.

Open Information Systems - Serbian Enterprises Standing and Perspectives

Slavoljub Milovanovic

Open information systems are based on client-server architecture which makes the environment of distributed data processing. In such an environment application may be developed on one platform, execute on the other one, while data may be stored on the third type of platform if it is necessary. Enterprises which implement open systems satisfy one very important requirement needed for potential improvement of those systems, and that is that present choice of hardware and software products does not limit future choice, when better and improved products appear. However, these systems implementation is not a simple one at all, and that is to be shown in this paper on the example of Serbian and foreign enterprises.

In the paper are given empirical researches results which refer to open systems concept implementation in Serbian enterprises. Practical instructions for this concept implementation are given in the paper as well. Researches show that the concept of open systems which are to be distributed, transparent and interoperable ones is not realized. Our enterprises information systems may be said to be nonflexible, nonmodulated and with limited possibility of combining and adapting various products of information technology made by a lot of producers. They are still directed towards classical data processing, while support to user cooperation, resources distribution and communication are completely neglected.

30 enterprises, mostly social ones (29 social and 1 private enterprise) are dealt with in this research. 20 enterprises are taken from the production sector and 10 from nonproduction. Producing enterprises are taken from different industries: food-processing industry, tobacco industry, textile industry, rubber industry, chemical industry, mechanical industry and electronic industry. As for non-producing enterprises 6 banks are involved in the research, 3 public enterprises and 1 commercial enterprise. Data are collected by personal interview method. It was a nonstructured interview with the information function executives, and it consisted of questions, which referred to hardware, software and cadre basis of information function, as well as data organizing in our enterprises.

Software development of Medical Image Archiving System

Antal Nagy, László Nyúl, and Zoltán Alexin

SZOTE-PACS has being developed since 1995. The goal of the system is to provide a modular, flexible PACS (Picture Archiving and Communication System) which is able to collect digital images of patient studies from different modalities and archive them in a central database. The end users can retrieve the requested images from the image database.

SZOTE-PACS is based on the DICOM standard which is a common file format and communication protocol in medical imaging. There are lots of different imaging modalities producing different image formats (TIFF, Interfile, ACR-NEMA, DICOM). Each modality is connected to the University Network. The old modalities which produce non-DICOM output are connected to converter stations. These stations and the newer modalities are directly connected to the main image server via NFS, FTP and DICOM protocol. The non-DICOM studies can be converted automatically into DICOM format. The image server can receive images both from the admitting stations and from the outside world via the Internet. The incoming images are stored in a temporary directory from where the server builds the studies into the Oracle database. This database stores the image data in a Patient, Study, Series, Image hierarchy. The images are stored in this database as a reference to the physical image file. The disk capacity of the server (26 GB) is enough to keep the studies for at least 15 days. After this period the images are removed from the server. The header data will remain in the database forever. In these two weeks the images can be archived to CD-ROM. End users can reach the image database from the viewing stations using the Oracle SQL server. On the viewing stations physicians are able to create educational material and to create local scientific databases for their work.

On the admitting stations and on the server the processes can be executed in an automatic way. It means that the automatic process takes into account the result of the previous process and the free disk spaces. Errors are logged into an information file. The executed commands are stored in a diary which helps to reconstruct the system events.

SZOTE-PACS is able to include study parameters from the Radiology Information System (RIS). The form of the RIS database is a DBase database. The admitting stations are connected to the RIS database via RIS gateway which receives SQL queries and forwards them to the RIS. The answer from the RIS goes through the RIS gateway to the admitting stations as well.

Built-in Scheduling for Protocol Design

Endre Németh

The real-life protocols are the smallest logical units of communication. The protocols are designed as software elements from cooperating processes and some kind of scheduling is necessary among the processes. My goal is to describe and solve this scheduling problems. First I translate the protocol problems to the language of mathematics from the point of view of scheduling. The result of this translation is similar to the job-shop problems, but it differs sharply from it. The main differences are the following:

- it contains stochastic elements
- it has inventories.

I attempt to maximize the number of processed messages applying the best scheduling method.

The messages can be considered as jobs. The operation is the processing of the messages. A communication protocol is composed of the following processes:

- receivers,
- transmitters,
- controller and
- timer.

The paths of the messages among the cooperating processes are determined by constraints specified as state-transition rules. The receivers and transmitters are coders and decoders, the timer is a counter. The controller is a set of state-transition machines containing the state-transition rules. The scheduling is considered among the cooperating processes.

The scheduling problem considered among the cooperating processes is solved under the following main assumptions:

- no two operations of the same message may be processed simultaneously,
- each operation , once started, must be completed before another operation may be started on that process
- each message has m distinct operation, one on each processor
- the incoming messages are specified as arcs of two different protocol trees

We apply a heuristic algorithm to solve the problem. The validity of the algorithm is checked by simulation of a mobile protocol.

Syntactical Analysis of Hungarian Sentences to Produce Prosodic Information for Speech Synthesis

Péter Olaszi

Linguistics consultant:
Ilona Koutny, Uniwersytet Adama Mickiewicza, Poznań, Poland
Speech synthesis advisors:
Gábor Olaszy, Hungarian Academy of Sciences, Linguistics Institute
Géza Németh, TUB DTT, Speech Research Laboratory

Telecommunication related speech research has traditions at the Department of Telecommunication and Telematics of the Technical University of Budapest. Good results were archived by the speech recognition groups lead by Klára Vicsi and Péter Tatai. The speech synthesis group coordinated by Géza Németh and Gábor Olaszy has been developing the Multivox multilingual speech synthesizer available for Hungarian, German, Italian, Portuguese and other languages. Our aim now is to produce more natural sounding synthesized speech.

One way to improve speech quality is to add prosodic information to the text. Prosody, ie. the base frequency, amplitude and rhythm changes, is generally not carried by the written text. The reader, recognizing the syntactic structures and the actual meaning of the text, indicates the focus, adds emphasis and pauses to the sentence. Providing prosodic control sequences for the synthesizer, it can parse this additional information and use it when forming the intonation. This results in more natural sounding speech. The control sequences can be added to the text either manually or automatically.

This paper discusses the problem of automatic prosodic information generation for Hungarian by means of syntactical analysis of the sentences. This process can be divided into two separate phases. In the first phase the syntactic analyzer performs an analysis on the sentence. The output of this module is a representation of the structure of the sentence: the predicate, object, subject and other parts of the sentence are identified. This information forms the input of the second phase. The prosody module performs two tasks: assigns stress to each word, and places pauses into the sentence, using phrase boundary information.

The operation of the syntactic analyzer is described in detail. Phrase detection is based on phrase pattern matching, using unification techniques. Phrase patterns are stored in a separate file — the rule matching engine is language independent. The structure of the sentence is determined using structure describing rules.

Stress and pause assignment is performed by the prosody module. The input of this module is the result of the syntactic analysis. Both syntactic and prosodic rules are programmed according to the directions of Ilona Koutny.

New issues in this work:

- A new representation of the syntactic structure is introduced. This representation can provide information for the prosody generator. The result of the syntactic analysis can also be used by any other natural language processing program.
- Syntactic analysis of Hungarian poses special issues. The major difficulties derive from the agglutinative character of the language. This also results in complex morphology of the words and unconstrained word order in the sentence.
- We developed a prosody generator using the output of the syntactic analyzer. This prosody generator can be used as a development tool for the speech synthesizer.

The prototype of the system is worked out for single-clause sentences of constrained topic. The sentences were taken from weather forecasts. The program is implemented in standard C.

Generation of JAVA classes from ASN.1 for BER transfer syntax

Viatcheslav V. Ostapenko

In order for communication to occur it is necessary to specify the type and format of the information to be exchanged. A general, distributed application protocols use very complex types of information. It is therefore useful to have a formal tool that permits the precise definition of the types of information exchanged. It is also useful to have a set of rules that specifies the format of the values for each of the types. Each instance or value of a type must be converted to this format before transmission. Abstract Syntax One (ASN.1) with its associated encoding rules is an example of a formal tool for specifying the type and transfer format of type values. ISO and ITU-T define it with its encoding rules as an external data representation language for communication of heterogeneous systems (ITU-T X.208, X.680). This tool has been used extensively with existing OSI specification protocols as well as many new applications.

External data representation languages support two components: facilities to describe data in terms of a set of application specific data type and facilities to define how values of these types are represented serially for a communication. The fires is termed abstract syntax and the second transfer syntax. In earlier representation languages these two concepts were combined. Current languages such as ASN.1 make a clear separation between them. This distinction permits multiple transfer syntaxes to exist. There was earlier the only one standard set of encoding rules for ASN.1 These are called Basic Encoding Rules (BER - ITU-T X.209, X.690). Now there are newer encoding rules, but this talk does not deal with them.

To use data structures described by ASN.1 they must be programmed in a common programming language. It would certainly be a great advantage to an application programmer if an automatic process existed which would convert the types in an abstract syntax into types used by a specific programming language. It would also be very useful if a set of encoding routines was generated that could be called by the programmer to convert a value of a type to and from particular transfer syntax. The biggest part of such translators is generate C source code with structures and encoding procedures for them. C language is considered portable language so source code can be used in different systems and platforms. But as practice shows there is a lot of incompatibilities between different compilers. Also a big number of different structure and procedures to process them gives a big possibility of making error by programmer and C type checking mechanism does not provide strong enough type checking.

In this talk we describes method of translation of ASN.1 data representation and BER rules into JAVA classes and methods to process data of that classes.

JAVA is a programming language environment suggested Sun Microsystems as common language for any platform. It is simple, object and network oriented, robust, secure, architecture neutral, high-performance, multithreaded, dynamic language. Java is designed to meet the challenges of application development in the context of heterogeneous network-wide distributed environments. Paramount among these challenges is secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform, and can be extended dynamically.

Object oriented structure of JAVA allows generation encoding methods for all classes with the same names inside classes, so each class will KNOW how to encode and decode data in it. Every constructed class including several other, may be also constructed, classes can easily encode them by calling encoding methods for each class. The only exclusion is a set of simple built in types. JAVA stream classes allows designing encoding methods to know how to store data to a stream and appropriate stream class will work with a hardware device or file. It simplifies programming process greatly.

Generation of JAVA classes from ASN.1 data description can be effectively used for programming Internet oriented applications. It is also useful for generation tests from TTCN (Tree and Tabular Combined Notation) that uses ASN.1 for specification of internal data types. Tests in JAVA will run in every system and there is no need to regenerate them for different platforms.

This talk describes method of generation only BER encoding rules in JAVA, but it can be easily modified to generate any other transfer syntax.

A New Approach to Neural Network Design

András Péter and Lehel Csató

The radial basis function (RBF) neural networks were proposed in the mid-80s for classification and approximation. These networks work with a single hidden layer in most of the cases. The basic idea behind them is that the function or the class is approximated by a combination of basis functions, what differs significantly from zero on a small bounded region. This permits the local tuning of the RBF networks.

The parameters of the basis functions are fixed in the basic version of the RBF networks, and the optimization is done on the level of the weights by that these functions are combined to obtain the approximation. In more advanced versions, it is permitted the modification of these parameters too.

Starting from the mid 80s, it was established that the RBF neural networks have the universal approximation property with respect to the continuous functions, and error bounds were calculated for these networks.

A newer approach is proposed in this paper with respect to the approximation properties of these networks. By this approach it is possible to estimate the number of the necessary hidden nodes and their internal parameters, and to evaluate how appropriate is the application of the RBF neural networks for the approximation of classes of functions.

The orthogonalization of the RBF neural networks in fact means that we look for an orthogonalization method of the Gaussian functions. The idea is that if we can find a proper orthogonalization then we can apply the theory of approximations with combinations of orthogonal units.

By orthogonalization of the set F of the Gaussian functions of form $f : R^n \to R, f(x) = e^{-\frac{||x-c||^2}{2r^2}}$, we mean that we to find an infinite subset $F_0 \subset F$, and an internal product, such that we have for all $f_1, f_2 \in F_0$ that

$$< f_1, f_2 >= \begin{cases} c_{f_1} \; if \; f_1 = f_2 \\ 0 \;\; if \; f_1 \neq f_2 \end{cases} .$$

We note the Hilbert space generated by this internal product by H. Then we can apply the theory of the orthogonal approximations within H. The paper presents the basic results in this respect.

We define the Gaussian spectrum of the goal function and we present a network design method based on the analysis of this spectrum. Using this theoretical network design framework we present applications to function approximation and time series prediction. We found in concordance with our expectations that the spectrum-based neural networks perform much better than those with randomly selected parameters.

Further work should be done in order to generalize the proposed approach to other types of neural networks.

The Correspondence Between Varieties of Automata and Semigroups

Tatjana Petković

For a finite alphabet X, a non-empty class of recognizable languages in the free monoid X^*, closed under Boolean operations, left and right quotients and inverse homomorphic images, is called a *variety of X-languages*, and by a *variety of languages* we mean a mapping $X \mapsto \mathcal{L}(X)$ which to each finite alphabet X associates some variety of X-languages $\mathcal{L}(X)$. On the other hand, automata with the input alphabet X are often considered as algebras of unary type indexed by X, so by an *variety* (resp. *pseudovariety*) *of X-automata* we mean an ordinary variety (resp. pseudovariety) of unary algebras of this type and by a *variety* (resp. *pseudovariety*) *of automata* we mean a mapping $X \mapsto \mathcal{A}(X)$ which to each alphabet X associates some variety (resp. pseudovariety) of X-automata $\mathcal{A}(X)$.

The well-known theorem of S. Eilenberg, proved in 1976, says that there is a bijective, order preserving correspondence $V \mapsto \mathcal{L}_V$ between pseudovarieties of monoids and varieties of languages. A similar correspondence between semigroups and languages in free semigroups was established by J. E. Pin in 1984. On the other hand, it was proved by M. Steinby in 1994 that there is a bijective, order preserving correspondence between pseudovarieties of monogenic automata and varieties of languages. Therefore, the remaining open problem is to study related correspondences between automata and semigroups, This is the main purpose of the present paper.

The main result that we prove here is the following general theorem:

Theorem 1 *There is a bijective, order preserving correspondence $V \mapsto \mathcal{A}_V$ between varieties of semigroups and regular varieties of automata.*

Here by a *regular variety of X-automata* we mean a variety determined by regular identities, that is of identities of the form $pu = pv$, where p is a variable and $u, v \in X^*$.

Similar correspondences between pseudovarieties and generalized varieties of semigroups and regular pseudovarieties and generalized varieties of automata will be also established. For a given cardinal κ we consider semigroups having not more than κ generators, called *κ-generated semigroups*, and classes of κ-generated semigroups closed under homomorphic images and κ-generated subdirect products, called *κ-varieties*, and we establish a correspondence between varieties of X-automata and κ-varieties of semigroups, where κ is the cardinality of X.

We also study non-regular varieties, pseudovarieties and generalized varieties of automata. The greatest non-regular generalized variety of automata is the class of all *directable automata*. Besides the "classical" characteristic semigroups, we introduce and study a new type of characteristic semigroups of directable automata, defined in terms of non-regular identities satisfied on them. We consider the problem of *regularization* of non-regular varieties of automata, studied previously by J. Płonka in 1982–95, E. Graczyńska in 1983, S. R. Kogalovskiĭ in 1991 and others, and we connect it with the concept of *localization* of varieties of automata, introduced by M. Steinby in 1994.

Moreover, we investigate some well-known generalized varieties of automata, consisting of directable, definite, reverse definite, generalized definite and nilpotent automata, and we introduce some their generalizations: generalized directable, trapped, one-trapped, locally directable, locally one-trapped, locally nilpotent and locally definite automata. These types of automata will be completely described in terms of their characteristic (transition) semigroups, and using several well-known decomposition methods, such as direct sum decompositions, subdirect and parallel decompositions and extensions of automata.

On Specialized Programming Language for Flexible Information Systems

Aleksandar Popovic, Zoran Budimac, and Mirjana Ivanovic

It is often the case that processing of some data classes = needs parameters. However, there are no easy solutions if processing = parameters must be expressed by algorithm (or program) and not by = "ordinary" data. In this paper a specialized programming language = (called PLES) that solves the problem is described. Programs of this = language are stored into the database together with the rest of the = data. Programs can be changed and executed without any support from = development team or development environment of the information system. = By using programs of this specialized programming language, information = systems can be more flexible and significantly easier to maintain. = Half-finished software components can be easily built as well.

Programming language PLES contains:

* several built-in simple data types (fixed point numbers, strings, = booleans, characters, etc.)

* two built-in structured data types (arrays and records)

* subroutines (procedures and functions)

* several control structures (if-then-else, while loop, for)

* a possibility to access "external" data (i.e., a data from the PLES = program environment: fields of a screen form or database tables). This = access is compatible with the access to ordinary PLES program variables.

Depending to which external data PLES program can access, PLES is = defined in two variants: PLES-Client and PLES-Server. PLES-Client = programs can access only screen form fields and can be executed at the = client side of an information system. PLES-Server programs can access = only database table fields and can be executed at the server side of an = information system.

PLES is implemented as two components:

* a compiler into a machine language of an abstract machine and

* an interpreter of machine language of that abstract machine

The choose solution is appropriate, because of the following reasons:

* It is easily portable.=20

* Abstract machine code can be easily and quickly transmitted through a=20

* Execution of abstract machine code is more efficient that a direct = interpretation of PLES programs.

Using Object Oriented Techniques at Implementing Compilers

Zoltán Porkoláb

Object Oriented Programming became well-accepted technology of Computer Science in the recent years. Object Oriented Methods of analysis and design are to standardize and supported by CASE tools. Programming languages - such C++, ADA95 and JAVA - are in everyday use helping the programmer to implement object oriented techniques with classes, using inheritance, exception handling, etc. In a few fields of Computer Science, however, Object Orientation is still not a mayor player. In specially, in the world of compilers, translators, etc. we can see mostly old-fashioned, not object oriented programs. This is perhaps originated in the event-driven attitude of using finite automaton.

In this article I try to explain another way of writing translators - the object oriented way. In this case the object structure of the program is driven from the internal data structures of translation. Lexical tokens are detected by lexer, and passed to the parser as an object. The parser is building a syntax tree using Predictive Descending Algorithm (Aho, Sethi, Ullman: Compilers. Addison-Wesley, 1986), recursion was eliminating with while loops. However this syntax tree is containing objects from a certain object hierarchy. Each node in the tree is an instance of a class corresponding to a certain operator-group. This way of typing the nodes of syntax tree makes late binding in effect at type checking and code generation.

Type checking makes function overloading possible. Rules of function argument matching are separated in distinct algorithms. A few case of polymorphic function needs special handling. Code generation also use virtual functions producing ready to compile JAVA source. Exception handling is used not only to break syntax analysis at error but also to generate warnings. Error and warning information is encapsulated in classes and can be caught.

This translator is a part of a real-word project, implemented in JAVA 1.1. In the project 4GL (SuperNOVA) source code is translated to JAVA (applet) source. The translator is validating the 4GL expressions and generate JAVA code. Since this 4GL is based on interpreter and allows run-time evaluation of expressions, the class structure of the translator is also a part of the applet. This way the translator is used in compiling time to check the most of the source and generate static JAVA code, but also activated in run-time if a certain expression is using run-time evaluation. The project has finished in the last year and now the translator is ready to use.

On Length of Directing Words of Automata

Antal Pukler

A finite automaton $\mathbf{A} = (A, X, \delta)$ is directable if there exist an input word q and a state b such that $aq = b$ for every state a of automaton. The word q is called directing word. In 1964 Cerny [1] conjectured that a directable automaton with n-state must have a directing word of length less or equal to $(n-1)^2$. Recently, this conjecture is proved in cases $n \leq 5$, for cyclic automata with prime number of states and for monotonic automata. There are some special cases of directable automata where the upper bound is less then $(n-1)^2$ (for example the bound is $n-1$ for the commutative, definite and nilpotent directable automata). In general case the conjecture is not proved, and the best known upper bound is $\mathbf{O}(n^3)$.

Let $A'x = \{ax | a \in A'\}$ and $A'x^{-1} = \{a | ax \in A'\} (A' \subseteq A)$. The symbol x is called idempotent if $Axx = x$, and x is a simple idempotent if there exists only one $a \in A$ such that $|\{a\}x^{-1}| > 1$. It has been proved by Rystsov [4] that a directable automata with simple idempotent and n states must have an input word with length less or equal to $2(n-1)^2$.

In our paper the directable automata having a simple idempotent is studied.

Let $\mathbf{A} = (A, X, \delta)$ be an automaton, x an input symbol and $A' = \{a_0, a_1, \dots, a_{m-1}\} \subseteq A$. A' is called an x-cycle if $a_i = a_{(i+1) \mod m}$ for every state $a_i \in A'$, the length of A' is m.

First we will assume that the directable automata has an input symbol x (without the simple idempotent) for which A is an x-cycle. We will prove that the minimal length of their directing words is not greater than $(n-1)^2$.

In the second case we assume that there exists a state a for which $\{a\}$ is an x-cycle and $A - \{a\}$ is an x-cycle too. In this case we can also prove that the minimal length of the directing words is not greater than $(n-1)^2$.

References

[1] J. Cerny, Poznamka k homogennym experimentom s konecnymy automatami, Maht. Fyz. Cas SAV, 14 (1964), pp. 208-215.

[2] D. Eppestein, Reset Words for Monotonic Automata, SIAM J. Comput., 19 (1990), pp. 500-510..

[3] B. Imreh, M. Stenby, Some Remarks on Directable Automata, Acta Cyb. Vol. 12. No 1 (1995) 23-35.

[4] J. Pin, Sur un cas particulier de la conjecture de Cerny, in Lecture Notes in Computer Science 62, Springer-Verlag, Berlin, 1978, pp. 345-352.

[5] I Rystsov, Reset Words for Automata with Simple Idempotents. ICALP'97. (1997).

An Analysis of Some Characteristics of Different Programming Paradigms

Zoran Putnik

It has been defined that: "A programming paradigm is a collection of conceptual patterns that together mold the design process and ultimately determine a program's structure [2]". Because of the fact that programs are usually executed on a Von Neumann style based computers, not best suited for programming styles different than procedural, it was not possible to correctly compare those paradigms for a long time.

With development of computers, systems allowing programming in different programming styles, or even combining different programming styles in a single program, became available. One of them is programming package *Leda*. Availability of such a system, leads to a set of interesting questions. Should we reconsider our opinions about "most efficient" programming style. Which one is the best? The "fast" one, the "readable" one, or the one which can be proven correct via mathematical formalisms.

At the beginning of development of programming paradigms, it was possible to recognize languages strictly connected to a certain paradigm. Example of *imperative* paradigm was *FORTRAN*, of *functional* paradigm was *LISP*, of *logic* paradigm was *PROLOG*. Later, it became more complicated. Natural need for combining the best features from different programming paradigms, induced programming languages that borrow from many paradigms and support more than one.

Along the way, some other programming paradigms emerged, more or less successful. Object-oriented, parallel (asynchronous or synchronous), transformational, form-based, dataflow, constraint, demonstrational and so on [2]. All of them can still be separated into two main different categories: *operational* or *demonstrational* approach, describing step-by-step how to construct a solution, and *definitional* approach, stating properties about solution, without describing how to compute it. In our analysis, imperative - as operational, and logical - as definitional paradigms are included. Functional paradigm, having characteristic of both categories is included as a fine transition from one to the other.

While discussion on readability, or correctness may be a matter of opinion, discussion on speed of programs may be checked rather agreeably. In order to gain correct and valuable results, several different kind of problems have to be included, best suited for both operational and definitional approach, i.e. for each of mentioned programming styles. This includes, for example: "Declarative" problem - mathematical problem that can be stated as a set of facts and rules, i.e. as a program written in a logic programming style, "Iterative" problem - mathematical problem, naturally suited for imperative style and procedural programming, that can still be programmed in both functional and logical manner, and "Recursive" problem - mathematical problem in which a new result is gained from a previous results, best suited for a functional programming style. All these problems are programmed in each paradigm and execution times are measured. My conclusion is: logical and procedural paradigms had almost the same execution times. Functional paradigm had slower execution time, but still very competitive one.

This paper is aimed to help in better understanding of some of the existing paradigms, oldest and most spreaded, and in changing our opinions about some "facts" about programming styles that may not be true. I presume that some kind of procedural programming (object-oriented, for example [1]) will be mostly used in the future, for which there are lots of reasons, beginning with the computer's architecture and programmers long-time habits. My main interest was to show that given a proper programming package - supporting different programming styles, a proper knowledge about the problem we have to solve - allowing combinations of programming styles and proper computer and software architecture - allowing execution of different programming paradigm, we would be able to achieve better, more efficient, more readable and mathematically correct programs.

References

[1] Rine, C.D., Bhargava, B., Object-Oriented Computing, Computer, October, 1992., pp. 6-10.

[2] Ambler, L.A., Burnett, M.M., Zimmerman, A.B., Operational Versus Definitional: A Perspective on Programming Paradigms, Computer, September, 1992., pp. 28-43.

Regular Grammar Model for Protocol Testing

Csaba V. Rotter

This paper presents a new approach on conformance test modelling. Protocol testing is the practical way to check correctness of protocol implementation with respect to their specification. TTCN (Tree and Tabular Combined Notation) is the standardised test notation for the description of OSI conformance tests. Fundamental principles of protocol conformance testing is a regular exchange of PDU 's (Protocol Data Unit) between the peer entities and ASP 's (Abstract Service Primitives) to communicate to the neighbouring layer. The communication dialog between the tester and IUT (Implementation Under Test) is regarded as an exchange of words with help of well defined regular grammar. According to this new theory of modelling all PDU-s, ASP-s and practically all events defined on TTCN are considered as a set of alphabet on a regular language.

A language (Formal Language) is a set of sentences and a sentence is a finite sequence, or string of elements, each of which is drawn from a finite vocabulary.

A language can be defined by a particular grammar as the set of sentences it generates. The grammar is an ordered quadruple

$G = (\Sigma, V, P, S)$

where:

Σ - set of terminal symbols (PDU's, ASP's,....)

V - set of non-terminal symbols (states of the IUT, states of Tester)

P - set of the production rules on the system (Rules of the comm. dialog)

S - is the starting component of the generation (Idle state of IUT)

If we can find a set of symbols from which the participants (the tester and the IUT) compose their sent sentences and we can construct the grammar containing the rules correctly generating these sentences we can get the formal description of testing. The steps for constructing such a grammar are:

- to determine the set of symbols sent by the tester to IUT and to determine the set of symbols sent by the IUT to tester. The union of these two sets forms the set of symbols characterising the message exchange they will form the set of terminals.

- to provide the non-terminals, which can be all of the IUT 's "state" where it can arrive after changing a dialog with the tester

- to construct the set of production rules for the grammar following the sequence of the dialog

Finally we attempt to construct such as grammar for a real test case for DECT testing and we focus to the problems which can appear in these situation and the advantages and disadvantage of this method.

Automatically Generated Icons for Documents of Electronic Libraries

Ádám Schmideg

Most documents on the internet assume users to have similar capabilities as those of computers. Keywords are listed, information is grouped in tree structures. Human mind however seems to work a different way. Everyone had an experience that he or she did not remember a sentence in a book but its location on the top of the page. Or he or she remembers a book's color and size but not its logical category in the library. The steady growth of information on the internet gave rise questions concerning document visualization. The emphasis however has been put on visualizing the structure of a collection of documents. The visual nature of human mind and memory suggests that separate documents also call for some kind of visualization or iconization. Since a text, even of moderate size, contains many ideas and thoughts, a detailed graphical representation would be very complex. Such a representation is likely to be less comprehensive than the original text. Thus, in order to get an emblematic icon for document

* one has to extract the content first
* find relatively simple icons for these concepts
* compose an image out of the basic icons

The first step is a recent and hot topic (though one approach was coined ICE - Intelligent Content Extraction) that is strongly related with the huge mass of information on the internet. The second step can be achieved using semantic knowledge to be able to select appropriate concepts as representatives of all possible thought within scope. And then it requires some artistic skill to depict these concepts.

We are mainly interested in the last step, i.e. in rules or algorithms of composing complex images from primitive ones. One approach is to use a template of placeholders for all images to be arranged. The simplest form of a template is a rectangular grid, so each image will be of the same size. More complicated patterns can be borrowed from heraldry where the constituting shapes are sometimes irregular, though the template as a whole has a delicate symmetry. Icons generated this way have very similar and easily predictable structure. In addition, they do not seem to capture some concepts in one but mostly fall apart into separate icons. Our method is to assign properties and special placeholders to each basic image. Then find a possible arrangement where all conditions are met - basic images fit into each other's placeholders, they are magnified to the extent allowed by their properties, etc.

This is a case of a complex packing problem that is difficult to solve in general. However, rules for placeholders and properties can be reduced to a subset. This new packing problem is much easier to solve and it still produces quite complex images closer to the ones generated manually. These icons are useful in many diverse areas,

* result set of a search on the Web
* visualization of heterogeneous databases
* collection of arbitrary hyperlinks
* intuitive user interfaces for internet applications

The generating method can be refined in many aspects, too.

Constructing Graphs with Given Eigenvalues and Angles

Dragan Stevanović

Introduction. The basic problem of spectral graph theory is how to construct all graphs with given eigenvalues. This problem is very difficult and for now there is no better method than to construct all graphs with given number of vertices and select those which have the given eigenvalues. If we also consider angles, beside eigenvalues, this problem becomes tractable. There are several known algorithms in the literature for the construction of graphs of special kind given its eigenvalues and angles. Cvetković gave such algorithm for trees [1], and for tree-like cubic graphs [4].

Previously, Cvetković [2] gave the method that uses eigenvalues and angles only to construct the graph which is the supergraph of all graphs with given eigenvalues and angles. Such supergraph is the *quasi-graph* in general case. If we also know the eigenvalues and angles of graph's complement, we can construct the *fuzzy image* of a graph, which enhances upon the quasi-graph. In the case of trees, that supergraph is the *quasi-bridge graph*, whose construction is much simpler than that of quasi-graph and fuzzy image.

In [3] Cvetković gave the lower bound on distance between vertices based on eigenvalues and angles of graph. In Section we give new lower bound, similar to this one and show that two are independent of each other.

Based on the lower bound on distance and the supergraph of all graphs with given eigenvalues and angles, in Section we give the branch & bound algorithm to construct all graphs with given eigenvalues and angles.

Notions Let G be the graph on n vertices with adjacency matrix A. Let the vectors $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ constitute the standard orthonormal basis for R^n. Then A has spectral decomposition $A = \mu_1 P_1 + \mu_2 P_2 + \cdots + \mu_m P_m$, where $\mu_1 > \mu_2 > \cdots > \mu_m$, and P_i represents the orthogonal projection of R^n onto $\mathcal{E}(\mu_i)$ (moreover $P_i^2 = P_i = P_i^T, i = 1, \dots, m$; and $P_i P_j = O, i \neq j$). The nonnegative quantities $\alpha_{ij} = \cos \beta_{ij}$, where β_{ij} is the angle between $\mathcal{E}(\mu_i)$ and $\mathbf{e}_j$, are called *angles* of G. Since P_i represents orthogonal projection of R^n onto $\mathcal{E}(\mu_i)$ we have $\alpha_{ij} = \|P_i \mathbf{e}_j\|$. The sequence α_{ij} $(j = 1, 2, \dots, n)$ is the ith *eigenvalue angle sequence*, α_{ij} $(i = 1, 2, \dots, m)$ is the jth *vertex angle sequence*. The *angle matrix* $\mathcal{A}$ of G is defined to be the matrix $\mathcal{A} = \|\alpha_{ij}\|_{m,n}$ provided its columns (i.e. the vertex angles sequences) are ordered lexicographically. The angle matrix is a graph invariant.

Lower bounds on distances. Let $d(j,k)$ be the distance between vertices j and k in G. Cvetković [3] gave the following lower bound on $d(j,k)$.

Theorem 2 (Cvetković [3]) *If* $g = \min\{s : \sum_{i=1}^{m} |\mu_i^s|\, \alpha_{ij}\, \alpha_{ik} \geq 1\}$, *then* $d(j,k) \geq g$.

The following theorem exploits the similar idea.

Theorem 3 *If* $g = \min\{s : \sum_{i=1}^{m} |\mu_i^{s+2}|\, \alpha_{ij}\, \alpha_{ik} \geq d_j + d_k + \delta_{s-1} - s\}$, *where* δ_{s-1} *is the sum of* $s-1$ *smallest degrees of vertices other than* j *and* k, *then* $d(j,k) \geq g$.

Example. For the tree shown in Fig. 2 from Theorem 2 we have $d(u,v) \geq 2$, while from Theorem 3 follows $d(u,v) \geq 3$. On the other hand, from Theorem 2 follows $d(w,v) \geq 3$, while Theorem 3 gives only that $d(w,v) \geq 1$. This shows that lower bounds given in these lemmas are independent of each other. In order to get better lower bound, one must then take the greater of the values given by these theorems. □

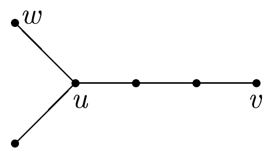


Figure 2: Graph from the Example of Section 2.

Quasi-bridge graphs. Cvetković in [2] gave the necessary condition for two vertices u and v to be joined by a bridge, and called it the *bridge condition*. He defined the *quasi-bridge graph* $QB(G)$ of the graph G as the graph with the same vertices as G, with two vertices adjacent if and only if they fulfill the bridge condition. If G is a tree, then we obviously have that G is a spanning tree of $QB(G)$.

The bridge condition is not sufficient for the existence of the bridge. On the other hand, there are trees for which the equality $QB(G) = G$ holds. Such examples are the stars S_n and the double stars $DS_{m,n}$ with $m \neq n$. From the statistical testing of random trees, the number of quasi-bridges of trees appears to be linearly bounded in the number of vertices. We believe that this fact holds for all trees.

Conjecture 4 *If $e(G)$ denotes the number of edges of graph G, then for trees hold $e(QB(G)) = O(e(G))$.*

Quasi-graphs and fuzzy images. Cvetković in [2] also gave the necessary condition for two vertices u and v to be adjacent, and called it the *edge condition*. He defined the *quasi-graph* $Q(G)$ of the graph G as the graph with the same vertices as G, with two vertices adjacent if and only if they fulfill the edge condition. Obviously, any graph is spanning subgraph of its quasi-graph.

Since the edge condition in $\overline{G}$ is a necessary condition for non-adjacency in G, it follows that any two distinct vertices of G are adjacent either in $Q(G)$ or in $Q(\overline{G})$. If they are adjacent in one and not adjacent in the other, then their status coincides with that in $Q(G)$. Cvetković [2] defined the *fuzzy image* $FI(G)$ as the graph with the same vertex set as G and two kinds of edges, solid and fuzzy. Vertices u and v of $FI(G)$ are

1) non-adjacent if they are non-adjacent in $Q(G)$ and adjacent in $Q(\overline{G})$;

2) joined by a solid edge if they are adjacent in $Q(G)$ and non-adjacent in $Q(\overline{G})$;

3) joined by a fuzzy edge if they are adjacent in both $Q(G)$ and $Q(\overline{G})$.

For the construction of $FI(G)$ one must know the eigenvalues and angles of $\overline{G}$. If G is regular and both G and $\overline{G}$ are connected then this information is already known from the eigenvalues and angles of G [5].

Except for small values of n (up to 4), it is very difficult to use the edge condition practically, so that the problem of its implementation still remains. Since the coefficients of the characteristic polynomials are integers, we can use the following much weaker theorem.

Theorem 5 *Let G be a graph with n vertices, and let uv be an edge of G. Then $P_{G-u}(n)P_{G-v}(n)$ is quadratic residue modulus $P_G(n)$ for every $n \in Z$.*

The Constructing Algorithm. The proposed algorithm is of branch & bound type. Before the algorithm enters the main loop, it determines the graph G^* that is the supergraph of all graphs with given eigenvalues and angles. Let G denote the fictious graph with given eigenvalues and angles, with vertex set $V(G)$ and edge set $E(G)$. In the case of trees we have $G^* = QB(G)$, in the case of regular connected graphs with connected complements $G^* = FI(G)$, while in other cases $G^* = Q(G)$. Then for each pair (u, v) of vertices, algorithm determines the lower bound $d_l(u, v)$ on distance in G between vertices u and v, based on Lemmas 2 and 3.

At level 0 of the algorithm we arbitrarily choose the vertex v_1, and pass to the level 1, where it enters the main loop. When we come at level i ($i \geq 1$) we have already constructed the subgraph G_i induced by vertices $v_1, v_2, \ldots, v_i$. This fact gives us the possibility to use the Interlacing theorem to check if we are on the good way with construction. Then we choose the vertex v_{i+1} from the set of remaining vertices, select its neighbors from the set $\{v_1, v_2, \ldots, v_i\}$, and pass to the next level, until G is constructed in whole. The vertex v_{i+1} is chosen such that the total number of neighborhoods of v_{i+1} that must be tried out is minimized.

References

[1] Cvetković D., *Constructing trees with given eigenvalues and angles*, Linear Algebra and Appl., 105(1988), 1-8

[2] Cvetković D., *Some possibilities of constructing graphs with given eigenvalues and angles*, Ars Combinatoria, 29A (1990), 179-187

[3] Cvetković D., *Some comments on the eigenspaces of graphs*, Publ. Inst. Math.(Beograd), 50(64) (1991), 24-32

[4] Cvetković D., *Some results on graph angles*, Rostock Math. Kolloq. 39 (1990), 74-88

[5] Cvetković D., Rowlinson P., *Further properties of graph angles*, Scientia (Valparaiso), 1(1988), 41-51

[6] Cvetković D., Rowlinson P., Simić S., *Eigenspaces of Graphs*, Encyclopedia of Mathematics and Its Applications, Vol. 66, Cambridge University Press, 1997

[7] Stevanović D., *Construction of graphs with given eigenvalues and angles*, M.Sc. Thesis, University of Niš, 1998

Packing of Equal Circles in a Square

Péter Gábor Szabó and Leocadio Gonzalez Casado

Several optimal packings of diverse shape objects are still open problems in geometry. The packing problem of this work is the optimal packing of n equal and non-overlapping circles in a square, where the radius of the circles are maximal. An equivalent problem is to locate n points in a square, where the minimal pairwise distance of n points is maximal. Let's denote this maximum distance by m. The problem can be described as [1]:

$$\frac{\begin{array}{c}(x_i - x_j)^2 + (y_i - y_j)^2 = s_{i,j}, \qquad \forall(1 \leq i < j \leq n) \\ 0 \leq x_i \leq 1 \qquad i = 1, \ldots, n \\ 0 \leq y_i \leq 1 \qquad i = 1, \ldots, n\end{array}}{\max_{x_i, x_j} \min_{(1 \leq i < j \leq n)} s_{i,j}}$$

where $s_{i,j}$ is the squares distance between the points i and j.

The packing problem have a long history in the mathematical literature, but this instance is only 38 years old. Leo Moser in 1960 was the first who studied this question. The proven optimal solutions given in the literature are only those up to $n = 27$ circles and that of $n = 36$. The approximations for larger configurations were done by numerical procedures. Good packings (the best known without a rigorous proof) over 50 are available for 52, 54-56, 60-62, 72 and the 78 [2, 3]. These good packings were found by repeated patterns method and minimization of the energy function. The most interesting story is for n=10. In the table we can see some people who worked with this case.

n	Year	Authors	m
10	1970	M. Goldberg	0.41666666
	1971	J. Schaer	0.41954209
	1979	K. Schlüter	0.42127954
	1987	R. Milano	0.42014346
	1989	G. Valette	0.42118970
	1989	B. Grünbaum	0.42124996
	1990	M. Grannell	0.42127954
	1990	J. Petris, N. Hungerbühler	0.42127954
	1990	M. Mollard, C. Payan	0.42127954
	1990	de Groot, R. Peikert, D. Würtz	0.42127954

Table 1: The n=10 case.

The method presented here is based on threshold accepting technique maximizing the minimum distance between one center of a circle and that of the others. This procedure is similar to simulated annealing but only not too bad configurations are accepted. The probability to find the optimum grows with the number of trials. To locate better positions of the center of the circles, modification of SASS (Simple Agent Stochastic Search) was applied [4]. This is the first stage of the algorithm. This stage stops when a reasonable number of searches were done. In the second stage more accurate solution is located from the best solution obtained in the previous stage. The accuracy is improved by overlapping a box over each obtained circle center and by the use of an interval branch-and-bound procedure [5, 6].

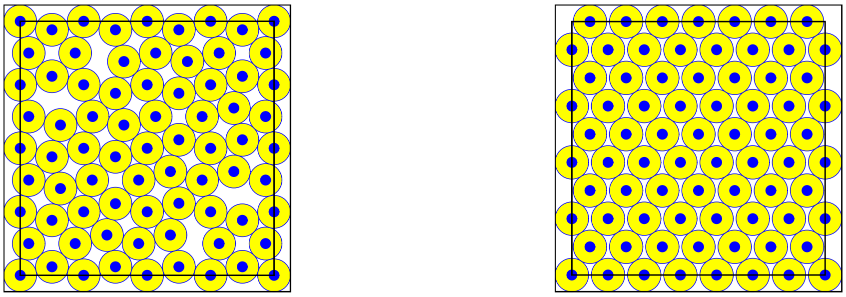
The interval branch-and-bound procedure can not be applied in all the cases because the larger the dimensionality the larger the number of combinatorial subproblems that make the problem intractable, and even reducing the number of combinations by taking into account symmetries will not help much [7, 8].

Beside the numerical results, we determined the exact m values for a lot of optimal packings.

n	m	n	m
2	$\sqrt{2}$	15	$(1+\sqrt{2}-\sqrt{3})/2$
3	$2\sqrt{2-\sqrt{3}}$	16	1/3
4	1	18	$\sqrt{13}/12$
5	$\sqrt{2}/2$	20	$(6-\sqrt{2})/16$
6	$\sqrt{13}/6$	23	$\sqrt{2-\sqrt{3}}/2$
7	$2(2-\sqrt{3})$	24	$4+2\sqrt{3}-\sqrt{26+15\sqrt{3}}$
8	$\sqrt{2-\sqrt{3}}$	25	1/4
9	1/2	27	$\sqrt{89}/40$
12	$\sqrt{34}/15$	36	$1/5$
14	$2(4-\sqrt{3})/13$		

Table 2: The exact values of the optimal packings.

Good packings are found a lot of new cases using a new pattern. For example in the Figure 1 we can see good packings for $n = 75$.



m=0.129409522 m=0.132089817

Figure 3: Packings of 75 circles

The good packings found with the discussed, as well as the best known packings. In Figure 2 we can see the m values as the function of the number of circles. The full circles mean the best results of the literature and the empty ones were found by the present procedure. Notice, that the new solutions fit well to the curve of the earlier values.

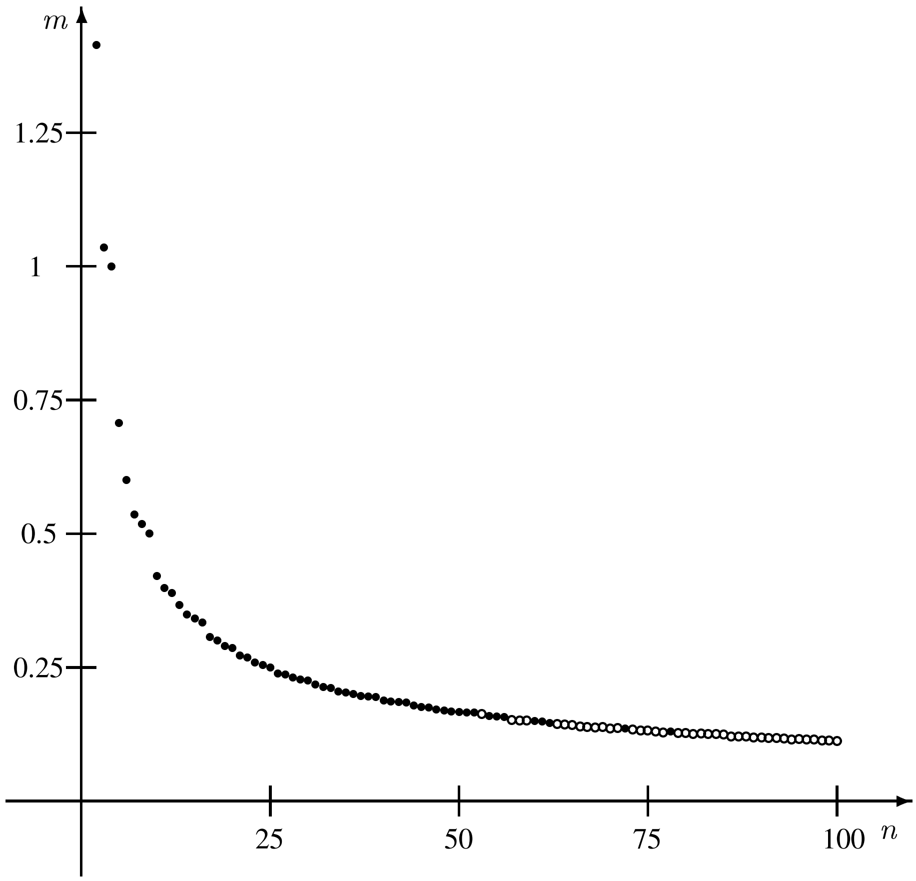


Figure 2: The maximum of the minimal distance as the function of the number of packed circles

References

[1] P.M. Pardalos, C.D. Maranas, and C.A. Floudas, New result in the packing of equal circles in a square. *Discrete Mathematics*, 142:187–293, 1995.

[2] K.J. Nurmela and P.R.J. Östergård, Packing up to 50 equal circles in a square. *Discrete & Computational Geometry*, 18:111–120, 1997.

[3] B.D. Lubachevsky and R.L. Graham, Repeated patterns of dense packings of equal disks in a square. *The Electronic Journal of Combinatorics*, 3:1–16, 1996.

[4] Francisco J. Solis and Roger J.-B. Wets, Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[5] L.G. Casado, I. García, and T. Csendes, Adaptive multisection in interval methods for global optimization, (submitted for publication).

[6] A.E. Csallner, T. Csendes, and M.Cs. Markót, Multisection in interval global optimization (submitted for publication).

[7] M. Monagan, R. Peikert, D. Würtz, and C. de Groot, Packing circles in a square: review and new results. In *15th IFIP Conference*, volume 180 of *System modelling and Optimization*, pages 45–54, 1991.

[8] K.J. Nurmela and P.R.J. Östergård, Optimal packings of equal circles in a square. In *8th International Conference on Graph Theory, Combinatorics, Algorithms and Applications*, 1996.

A Formal Approach to Design Patterns

Péter Szász

The term 'design patterns' has been established in the lexicon of software design by Gamma, Helm, Johnson, and Vlissides' pioneering book: 'Design Patterns: Elements of Reusable Object- Oriented Software'. A design pattern is a recurring architectural theme that provides a solution to common design problems within a particular context. Each design pattern systematically names, explains, and evaluates an important design decision in object- oriented systems. It identifies the participating classes and instances, their roles and collaborations, and the distribution of responsibilities. Probably every experienced object-oriented designer uses these patterns in some form (mostly instinctively), but the well-documented design patterns clarify the decisive situations, and predict the consequences.

A formalization would give a more stable basis for these useful tools, it helps classifying the patterns and analyzing its properties. An algebraic description is very useful for code generating systems or CASE tools, also.

We try to work out a formal approach, which can give a better understanding of the design patterns, revealing the regularities and differences in the structures of the most frequently used patterns. Nevertheless our intention is not to loose the connection with the practical applicability.

Multidimensional or Relational How to Organize an On-Line Analytical Processing Database?

István Szépkúti

In the past few years, the number of On-Line Analytical Processing (OLAP) applications increased quickly. These applications use two significantly different database structures: multidimensional and relational. One can show that the traditional model of relational databases cannot differentiate between these two structures. Another model is necessary to make the most important differences visible.

One of these differences is the speed of the system. It can be proven that the multidimensional database organization results in shorter response times. And it is crucial, since a manager may become impatient, if he or she has to wait say more than twenty seconds for the next screen.

On the other hand, here as well as in many other cases, we have to pay for the speed with a bigger database size. Why does the size of multidimensional databases grow so quickly? The reason is the sparsity of data: The multidimensional matrix contains many empty cells. Efficient handling of sparse matrices is indispensable in an OLAP application. One way to handle sparsity is to take the structure closer to the relational one. Thus the database size decreases, while the application gets slower. Therefore, other methods are needed to eliminate the empty cells from the matrix.

This paper deals with the comparison of the two database structures and the limits of their usage.

Efficient Implementation of Morphological and Local Neighbourhood Operations

Attila Tanács and Kálmán Palágyi

Mathematical morphology offers a powerful approach to numerous image processing problems. It is useful in the representation and description of region shape (such as boundaries, skeletons, and the convex hull) and in pre- or post-processing (such as filtering, thinning and pruning).

Morphological operations can be extended to gray–scale images, but our attention is focussed on binary images whose components are elements of $\mathbb{Z}^2$ or $\mathbb{Z}^3$. A digital binary image containing finitely many object points can be stored in a binary array in which 1's represent object points and 0's correspond to the background and the holes of the image.

A complex image analysis problem often involves the concatenation of several low–level morphological operations including dilation, erosion, hit–or–miss transformation, and point–wise logical operations. It is important to implement such low–level operations efficiently and to solve their concatenation as well.

There is a more general class of image operations called *local neighbourhood binary operations*. An image operation belongs to this class if and only if the value of every point in the destination image is a function of a small (say 3×3 in 2D and $3 \times 3 \times 3$ in 3D) neighbourhood of that point in the source image.

Local neighbourhood operations can be described by Boole–functions: a point is to be inverted if and only if the corresponding Boole–function is true for its neighbourhood. Any Boole–function can be replaced by a set of single-template predicates. A template is a small binary array (corresponding to the investigated neighbourhood) for defining a predicate: where a point satisfies the predicate if and only if the template matches its neighbourhood, where each 1 template element matches 1 point and each 0 template element matches 0 point. Note that no reflection or rotation is allowed in matching the template to the neighbourhood of the given point. In order to reduce the number of templates, templates can contain "don't care" elements, too. They match either 0's or 1's. It is easy to see that a template containing k "don't care" elements can be replaced by 2^k binary templates.

We introduce a new type of description that is called *generalized template matching*. It is a combination of Boole–formulas and templates. We have shown that any local neighbourhood operation can be described by this method.

The local dependence must be evaluated for every points of the image which is rather slow for large images. A general way for speeding up that kind of processes is using a lookup–tree: for all the possible neighbourhood configurations is evaluated only once and the the given values are stored in a binary tree. For n variables the tree has 2^n leaves. A common 3D operation can require more than 20 variables and in this case it is not efficient to store the whole tree thus tree–compression techniques are necessary. In order to reduce the size of trees, two methods are applied: *identical–subtree–compression* and *identical–leaves–compression*.

A special *programming language interpreter* is also proposed and developed for defining complex local neighbourhood operations. The graphical user interface provides easy and uniform way for giving and executing these operations.

There are some systems (e.g., KHOROS, AVS) that allow users to dynamically connect low–level operations or software modules to create data flow graphs for scientific computation. There are, however, some deficiencies of those systems which limit their usefulness (data flow graphs must be special, e.g. circle–free). Our programming language interpreter differs from the above systems in two regards: creating low–level operations does not need any programming; the structure of the complex operation is not limited (e.g., loops can be applied).

An example is to illustrate the simplicity and the efficiency of our approach. A 6–subiteration 3D thinning algorithm is given by the proposed language. The result of this morphological operation can be seen in Fig. 4.

```
rem 6-phase 3D thinning

picture A, B;
operation t1, t2, t3, t4, t5, t6;
int i;

assign(t1, "6thu.for" );
assign(t2, "6thd.for" );
assign(t3, "6thn.for" );
assign(t4, "6ths.for" );
assign(t5, "6the.for" );
assign(t6, "6thw.for" );

A=GetActivePicture();
B=CreateCompatiblePicture(A);
CopyPicture(B,A);
DisplayPicture(B);

InfobarText("6-subiteration 3D Thinning") ;
i=0;
while ( exec(B,B,t1 ) or exec(B,B,t2 ) or exec(B,B,t3 ) or
        exec(B,B,t4 ) or exec(B,B,t5 ) or exec(B,B,t6 )    )
{
   i=i+1;
   InfobarText("Number of itarations: %d", i);
}

DisplayPicture(B);
end;
```

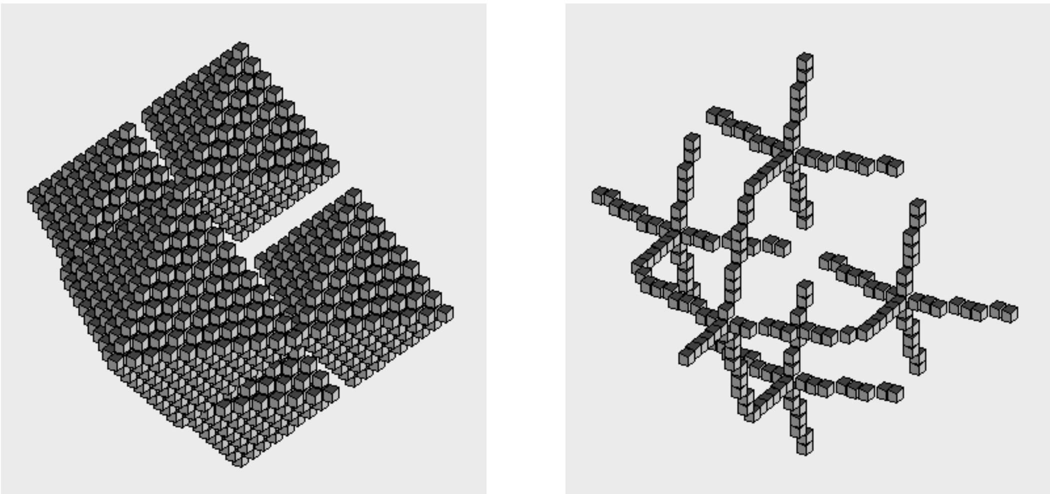


Figure 4: Example of 3D thinning. The original synthetic object (left); its medial lines (right). (Small cubes represent object points.)

Generation MSC and TTCN Descriptions for ISUP Supplementary Devices

Alexandr Tchanin

For improving ISDN quality and reliability, number of developed ISDN services increases progressively, and this process seems to continue for along time. As ISDN supports layered system architecture, these services are developed and implemented in bounds of ISDN User Part (ISUP) signalling protocol.

ISDN services within ISUP protocol are produced by European Telecommunication Standard Institute (ETSI) and realised in form of separate standards, that have common protocol specification, what makes their implementations easier. At current development and standardisation of ISUP basic services are actually finished, and now standardisation process is going on in frame of ISUP supplementary services. We chose 'Completion of calls to busy subscribers' (CCBS) ISUP supplementary service for generation descriptions.

Protocol engineering is the whole life-style of a protocol including the requirement specification, formal specification, validation and conformance testing so as in case of ISUP protocol. Conformance testing is a well-establishment testing methodology based on the multipart ISO 9646 international standard. In practice, its role is to increase confidence that a protocol conforms to the requirements stated in specification and to reduce the risk of malfunctioning when the protocol is implemented.

Protocol engineering is supported by Formal Description Techniques (FDTs). FDTs are used in requirement specification, formal specification, validation and testing, so as in case of ISUP protocol. Specification and Description Language (SDL), Abstract Syntax Notation One (ASN.1), Message Sequence Chart (MSC) and Tree and Tabular Combined Notation (TTCN) are the standardised language of ITU. MSC is a frequently used technique in requirement specification, SDL and ASN.1 in formal specification step of protocol engineering, TTCN is the notation defined by ISO, and used for specification of abstract test cases generated from formal specification in conformance testing. When implemented on a test tool, test cases assess a protocol. We used MSC and TTCN for description of ISUP CCBS.

TTCN describes the behaviour of the tester as a tree of events and actions, leading to verdict assignments (at the leaves).

A set of MSCs covers system behaviour since each MSC diagram represent one scenario of either a typical or an exceptional exchange of message between parts. MSC - as we have above mentioned - is a good tool for requirement specification, however it can be used for visualisation of a test case. To understand a test case behaviour in time is easier representing messages exchanged between tester and implementation.

Our experiences in this area have shown, that test specification should be generated semi-automatically in the appropriate language (e.g. TTCN) from protocol specification written in formal languages (e.g. MSC, ASN.1).

Hybrid Modelling and Reasoning in Measuring Systems

Gyula Tömösy

Measuring systems are a special form of information systems. The aim of their design is to extract some important information about the measured entities. The information flow begins at the sensors, then the measurement information is carried by complex signals to the information processing components, and to the user. Identifying or even detecting faults in such systems requires well based knowledge about the goal and the functionality of the measurement equipment. Traditional model-based diagnostic approaches cannot cope well with this problem, because of the complexity of the components of such systems. This paper introduces a novel approach to the modelling and diagnosing of measuring systems.

The first question to be answered is how to reason about the flow of information in measuring systems, i.e. about analog signals and systems. Planning the measurement provides high level knowledge about the role of the components in the measurement set-up and this knowledge should be exploited. Traditional theory of signals and systems provide a way to describe relations between parts in form of complicated mathematical formulas, e.g. convolution. Information about the measurement system is represented partly numerically and partly symbolically. The proposed approach introduces a multilevel abstract modelling scheme, where reasoning is possible at different levels of abstraction, using all the information available.

At the highest level of abstraction a functional description of the measuring system can be formulated. It describes certain teleological information about the system, introducing subtasks done by the components. At in-between levels qualitative knowledge can be used to describe the signals. Symbolical reasoning is still possible at this level. The lowest level are pure numerical signals.

The important point of the investigation is the reasoning about signal flow. Measurement takes place at numerical level, but reasoning is usually confined to symbolic information. Generally neither of the representation levels is fully available. Abstraction and refining support the transformation between representation levels. Several diagnostic reasoning strategies can be applied to this scheme depending on the completeness and presence of different kinds of description.

A central issue of the approach is thus knowledge representation. There is a need to have a formal way of symbolically reasoning about signals and systems, that will respect the mathematics of the system theory. This reasoning approach has several requirements to fulfill: (1) It has to ensure that the mathematical foundations wont be violated. (2) It has to grasp the essence of signals: enough, but not too detailed information has to be expressed. (3) Effective reasoning has to be supported by the representation.

The paper discusses the main problems of the topic, mainly concentrating on the problems of reasoning. The proposed system aim to provide a general representational method, that supports several fields of measuring, including e.g. measurement planning.

A Filter to Avoid the Aliasing Problem

Vesna Veličković

The aliasing problem appears when we display a mathematical object on the screen. It is interesting, the same problem not appears when we show a real-world scene. In this paper we consider the difference between the production of this two kinds of pictures and explain one way to avoid the aliasing problem.

When we want to display a mathematical object on the screen, we define a mathematical function that represents the desired object. Then, we take the samples of that function with which we evaluate the pixel values. In the case of a real-world scene, we record it by camera, then scan the photo (what is a sampling process again), and produce the pixel values. The main difference between that two processes is that we sample the different kinds of functions. The first ones **does not**, and second ones **does** satisfy the sampling theorem condition. An idea is to filter the function that represents the mathematical object **before sampling**, and make it to satisfy that condition.

Unfortunately, none of the well-known filter can be used. The digital filters cannot be used because they include the sampling process into themselves. The filters with infinite impulse response cannot be used because the filtering process will take very long time.

In this paper we explain the properties (defined in [1]) of a filter that can be used for this purpose. The filter properties in the frequency domain are chosen such that the components not satisfying the sampling theorem condition are suppressed enough and the others are used as much as they can be. The filter properties in the spatial domain are chosen such that enable efficient filtering, the sharp edges and invariation of the picture on rotation. Also, the filter should prevent a fine structure that doesn't exist on the desired scene.

In this paper it is explained a design of the filter. For the beginning, the design is done in a one-dimensional case by approximation with the orthogonal functions. It is explained why it is chosen such impulse response shape of the one-dimensional filter. It is shown the designed filter and its properties. Then it is explained the design of a two-dimensional filter from obtained one-dimensional one. It is shown the obtained two-dimensional filter and its properties. They are compared with the desired ones.

An experiment with filtering of the circles is explained.

References

[1] S. Hristov, M. Stanković, V. Veličković, "Exact" display of objects with real valued positions and dimensions, FILOMAT (Niš) 9:2 (1995), 251–259

Using Java and Erlang in Protocol Testing

Zsolt Werner

Even though there are established protocol testing methods and applications on mini computers (Sun), the Intel-based PCs lack any commercial application that would make it possible to either edit TTCN/MSC/etc... code on PCs or cross-compile code written in either language to any widely used programming language.

Both Erlang (http://www.ericsson.se/erlang) and Java (http://www.javasoft.com) are catching on pretty rapidly as two powerful new languages with a lot of potential, there is a significant problem in dealing with these languages, namely, the lack of protocol testing applications written in these languages.

The two languages can interface to native methods written in the other language. As both Erlang and Java offer advanced UI components that communicate with the end user, it's very easy to write Java / Erlang-based applications that can be used to edit TTCN/MSC code, print their class hierarchy, and all this in a highly portable form - the code runs on low-end PCs under Windows95/Linux, while for example, Proconsul, one of the leading products in this area, only runs on Sun workstations, and the same applies to most of the commercial applications devoted to this subject.

The goal for this paper is to introduce you to Java's/Erlang's UI (User Interface) programming and interfacing between the two languages - using the native methods written in the other language. If you are new to Java, you may not know what native methods are, and even if you are an experienced Java developer, you may not have had a reason to learn more about native methods. At the conclusion of this section you should have a better understanding of what native methods are, when and why you may want to use them, and the consequences of using them. You should also have a basic understanding of how native methods work. Of course, actual programming is not covered in the paper, it is not a tutorial, only our work is covered and ready-to-use result is presented.

List of Participants

Antic, Dragan: Faculty of Electronic Engineering, Beogradska 14, E-mail: dantic@elfak.ni.ac.yu

Areik, Adalla Khalil: E-mail: areik@ttt-202.ttt.bme.hu

Badjonski, Mihal: University of Novi Sad, Faculty of Science, Institute of Mathematics, Trg D. Obradovica 4, 21000 Novi Sad, Yugoslavia, E-mail: mihal@unsim.ns.ac.yu

Bak, Attila: Technical University of Budapest, Budapest, Hungary, E-mail: baki@twister.sch.bme.hu

Balogh, János: Institute of Informatics, József Attila University, Szeged, Hungary, E-mail: balogh@inf.u-szeged.hu

Bartha, Miklós: Department of Computer Science Memorial University of Newfoundland St. John's, NF, Canada, A1B 3X5, E-mail: bartha@cs.mun.ca

Battyányi, Péter: E-mail: battp@math.klte.hu

Benczúr, András: Eotvos Lorand University, Department of Information Systems 1088 Budapest, Muzeum krt. 6-8, E-mail: abenczur@ludens.elte.hu

Bernátsky, László: Department of Computer Science, József Attila University, P.O.B. 652 H-6701 Szeged, HUNGARY, E-mail: benny@inf.u-szeged.hu

Békési, József: Juhász Gyula Teachers Training College, Department of Computer Science, Szeged, E-mail: bekesi@cs.jgytf.u-szeged.hu

Berzsenyi, Miklós: Technical University of Budapest, E-mail: bermik@sun.aut.bme.hu

Boda, Károly István: Institute of Mathematics and Informatics, KLTE Debrecen 4010 Debrecen, Egyetem tér 1. Tel: (52) 316-666; (52) 416-857, E-mail: bodai@math.klte.hu

Bohus, Mihály: Department of Applied Informatics József Attila University, H–6701 Szeged P.O.Box 652, Hungary, E-mail: bohus@inf.u-szeged.hu

Casado, Leocadio G.: Dpto. Arquitectura de Computadores y Electrónica, Universidad de Almería 04120-Almería, Spain , E-mail: leo@miro.ualm.es

Ciric, Miroslav: University of Nis, P.O. Box 91, 18000 Nis, Yugoslavia, E-mail: mciric@archimed.filfak.ni.ac.yu

Csaba, Béla: Dept. of Computer Science Rutgers University, NJ, USA, E-mail: bcsaba@paul.rutgers.edu

Csallner, András Erik: Department of Computer Science, Juhász Gyula Teachers Training College 6720 Szeged, Boldogasszony sgt. 6, Hungary , E-mail: csallner@jgytf.u-szeged.hu

Csató, Lehel: Department of Computer Sciences, Eötvös Loránd University, Budapest H-1088 Budapest, Muzeum krt. 6-8, Hungary, E-mail: csato@zuzmo.iki.kfki.hu

Csendes, Tibor: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: csendes@inf.u-szeged.hu

Csima, Judit: Computer and Automation Research Institute, Hungarian Academy of Sciences, H-1111 Budapest Kende u. 13-17., E-mail: csima@luna.aszi.sztaki.hu

Csizmazia, Balázs: Eotvos Loránd University, Department of Information Systems 1088 Budapest, Múzeum krt. 6-8, E-mail: csb@ullman.inf.elte.hu

Devedzic, Vladan: FON - School of Business Administration, University of Belgrade, Jove Ilica 154, 11000 Belgrade, Yugoslavia,

Dimitrijevic, Sasa: Faculty of Electronic Engineering, Beogradska 14, 18000 Nis, Yugoslavia, E-mail: dimitrij@elfak.ni.ac.yu

Ekárt, Anikó: Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest 1518, POB. 63 , E-mail: ekart@sztaki.hu

Farkas, Levente: Department of Applied Informatics, József Attila University, H–6701 Szeged P.O. Box 652, Hungary, E-mail: lfarkas@ohsh.u-szeged.hu

Fülöp, Endre: E-mail: ulo@evt.bme.hu

Gál, Richárd: E-mail: galr@sch.bme.hu

Galambos, Gábor: Juhász Gyula Teachers Training College, Department of Computer Science, Szeged, E-mail: galambos@jgytf.u-szeged.hu

Gecse, Roland: E-mail: roland.gecse@lt.eth.ericsson.se

Gécseg, Ferenc: Department of Computer Science, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: gecseg@inf.u-szeged.hu

Gulyás, László: Eötvös Loránd University, H-1088 Budapest, Muzeum krt. 6-8, E-mail: gulya@dennis.inf.elte.hu

Gyenizse, Pál: Department of Computer Science, József Attila University, H–6701 Szeged P.O. Box 652, Hungary, E-mail: gyenizse@inf.u-szeged.hu

Hajdu, András: Lajos Kossuth University 4010, Debrecen P.O. Box 12, Hungary, E-mail: hajdua@tigris.klte.hu

Hajnal, Péter: Department of Set Theory and Mathematical Logic, József Attila University, Szeged, Hungary , E-mail: hajnal@math.u-szeged.hu

Hakonen, Harri: University of Turku, Computer Science Lemminkäisenkatu 14 A SF-20520 Turku, Finland , E-mail: hat@cs.utu.fi

Halmai, Csongor: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: halmai@sol.cc.u-szeged.hu

Holló, Csaba: Department of Informatics, József Attila University, Árpád tér 2, H-6720 Szeged, Hungary, E-mail: chollo@inf.u-szeged.hu

Hornung, Tamás: Department of Applied Informatics József Attila University, H–6701 Szeged P.O.Box 652, Hungary, E-mail: hornung@pluto.dfmk.hu

Imreh, Csanád: Department of Informatics, József Attila University, Árpád tér 2, H-6720 Szeged, Hungary, E-mail: csanad@inf.u-szeged.hu

Imreh, Balázs: Department of Computer Science, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: imreh@inf.u-szeged.hu

Jankovic, Dragan: Faculty of Electronic Engineering, University of Niš, Yugoslavia, E-mail: gaga@europa.elfak.ni.ac.yu

Janvári, Gusztáv: E-mail: janvari@dragon.klte.hu

Järvi, Jaakko: Turku Centre for Computer Science, Lemminkäisenkatu 14 A, FIN-20520 TURKU, FINLAND, E-mail: jaakko.jarvi@cs.utu.fi

Järvinen, Jouni: Turku Center for Computer Science TUCS, Lemminkäisenkatu 14, FIN-20520 TURKU, Finland, E-mail: jjarvine@cs.utu.fi

Jerinic, Ljubomir: Institute of Mathematics, University of Novy Sad, Trg Dositeja Obradovica 4, 21000 Novi Sad, Yugoslavia, E-mail: jerinic@uns.ns.ac.yu

Johnsson, Mika: Department of Mathematical Sciences and Turku Center for Computer Science (TUCS), University of Turku, FIN-20014 Turku, Finland, E-mail: johnsson@lena.cs.utu.fi

Kallós, Gábor: Széchenyi István College, Department of Computer Science, H-9026 Győr, Hédervári út 3. Hungary, E-mail: kallos@rs1.szif.hu

Kalmár, Zsolt: Department of Informatics, József Attila University, H–6701 Szeged P.O. Box 652, Hungary, E-mail: kalmar@mindmaker.kfkipark.hu

Karsai, János: SZOTE Orvosi Informatika Intézet, Szeged, Hungary, E-mail: karsai@dmi.szote.u-szeged.hu

Kincses, Zoltán: Lóránd Eötvös s University of Science Dept. of General Computer Science, E-mail: kincses@ludens.elte.hu

Kocsor, András: Institute of Informatics, József Attila University Árpád tér 2, H-6720 Szeged, Hungary, E-mail: kocsor@inf.u-szeged.hu

Kókai, Gabriella: Institute of Informatics, József Attila University Árpád tér 2, H-6720 Szeged, Hungary, E-mail: kokai@informatik.uni-erlangen.de, kokai@sol.cc.u-szeged.hu

Kovács, Emőd: Károly Eszterházy Teachers' Training School Institute of Mathematics and Informatics Leányka str. 4-6. , H-3300 Eger, E-mail: emod@ektf.h

Krész, Miklós: Department of Computer Science Memorial University of Newfoundland St. John's, NF, Canada, A1B 3X5, E-mail: kresz@jgytf.u-szeged.hu

Krtolica, Predrag: University of Niš, Faculty of Philosophy, Ćirila i Metodija 2, P. O. Box 91, 18000 Niš, Yugoslavia, E-mail: krca@archimed.filfak.ni.ac.yu

Kruzslicz, Ferenc: Janus Pannonius University H-7632 Pécs, Eszék u. 5 (72) 447-694, E-mail: kruzslic@ktk.jpte.hu

Lakatos, Attila: Eotvos Lorant University, H-1088 Budapest, Muzeum krt. 6-8., E-mail: kispitye@heureka.inf.elte.hu

Lehtinen, Joonas: Turku Centre for Computer Science TUCS, University of Turku, Lemminkäisenkatu 14 A, 20520 Turku, FINLAND, E-mail: jole@jole.fi

Lodha, Sachin P.: E-mail: lodha@paul.rutgers.edu

Lomic, Marijana: Faculty of Science, Institute of Mathematics, University of Novi Sad, E-mail: ilomic@unsim.ns.ac.yu

Magyar, Gábor: Turku Centre for Computer Science (TUCS) and Department of Computer Science, University of Turku, Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland, E-mail: magyar@cs.utu.fi

Malek, Mazen: E-mail: mazen.malek@lt.eth.ericsson.se

Markót, Mihály Csaba: Institute of Informatics, József Attila University, 6720 Szeged, Árpád tér 2, Hungary, E-mail: h430810@stud.u-szeged.hu

Maróti, György: E-mail: maroti@zenon.hu

Martinez-Ortigosa, Pilar: Dpto. Arquitectura de Computadores y Electrónica. Universidad de Almería. 04120-Almería. Spain., E-mail: pilar@iron.ualm.es

Martonossy, László: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: lmarton@sol.inf.u-szeged.hu

Máté, Eörs: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: mate@inf.u-szeged.hu

Micsik, András: MTA SZTAKI, 1111 Bp. Lagymanyosi u. 11, E-mail: micsik@sztaki.hu

Milovanovic, Slavoljub: E-mail: smilovan@orion.eknfak.ni.ac.yu

Nagy, Antal: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: nagya@inf.u-szeged.hu

Németh, Endre: MTA SZTAKI, 1111, Kende ut. 13-17, Budapest, Hungary, E-mail: nendre@cs.elte.hu, nemeth@cs.elte.hu

Németh, Zoltán: Institute of Mathematics and Informatics, KLTE Debrecen, E-mail: nemeth@math.klte.hu

Nikovits, Tibor: E-mail: nikovits@ullman.inf.elte.hu

Nyúl, László: Department of Applied Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: nyul@inf.u-szeged.hu

Olaszi, Péter: Technical University of Budapest, Department of Telecommunication and Telematics, Speech Research Laboratory, 1111 Budapest, Sztoczek u. 2. Hungary, E-mail: olaszi@twister.sch.bme.hu

Ostapenko, Viatcheslav V.: MTA SZTAKI, 1111, Kende ut. 13-17, Budapest, Hungary, E-mail: sl@proto.sztaki.hu

Péter, András: Department of Informatics and Mathematics, Babeş-Bolyai University Cluj-Napoca RO-3400 Cluj-Napoca, Kogãlniceanu str. 1, Romania, E-mail: peter@civitas.org.soroscj.ro

Petkovic, Tatjana: University of Niš, Faculty of Philosophy, Department of Mathematics Ćirila i Metodija 2, P. O. Box 91, 18000 Niš, Yugoslavia, E-mail: tanjapet@archimed.filfak.ni.ac.yu

Popovic, Aleksandar: University of Novi Sad, Faculty of Science, Institute of Mathematics , Trg D. Obradovica 4, 21000 Novi Sad, Yugoslavia, E-mail: ror@unsim.ns.ac.yu

Porkoláb, Zoltán: Eötvös Loránd University, H-1088 Budapest, Múzeum krt. 6-8, E-mail: gsd@caesar.elte.hu

Pukler, Antal: Széchenyi István College, Department of Computer Science H-9026 Győr, Hédervári út 3. Hungary, E-mail: pukler@szif.hu

Putnik, Zoran: Institute of Mathematics, Faculty of Science, University of Novi Sad, E-mail: putnik@unsim.ns.ac.yu

Radovic, Ognjen: E-mail: ogi@orion.eknfak.ni.ac.yu

Raita, Timo: University of Turku, Computer Science Lemminkäisenkatu 14 A SF-20520 Turku, Finland, E-mail: raita@cs.utu.fi

Rotter, Vilmos Csaba: Technical University of Budapest Telecommunications and Telematics Department, E-mail: rotter@sztaki.hu

Schmideg, Ádám: Technical University of Budapest, E-mail: schadam@bme-tel.ttt.bme.hu

Schrettner, Lajos: Department of Computer Science, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: schrettner@inf.u-szeged.hu

Sebő, Mariann: Institute of Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: sebo@inf.u-szeged.hu

Stankovic, Milanka: E-mail: mikica@o2edn.edn.ni.ac.yu

Stevanovic, Dragan: University of Niš, Department of Mathematics, Faculty of Philosophy, Ćirila i Metodija 2, P. O. Box 91, 18000 Niš, Yugoslavia, E-mail: dragance@ban.junis.ni.ac.yu

Stoerr, Hans-Peter: E-mail: haps@inf.tu-dresden.de

Szabad, Tamás: E-mail: szabadt@twister.sch.bme.hu

Szabó, Péter Gábor: Department of Applied Informatics, József Attila University, H–6701 Szeged P.O.Box 652, Hungary, E-mail: pszabo@inf.u-szeged.hu

Szász, Péter: Department of Applied Informatics József Attila University, H–6701 Szeged P.O.Box 652, Hungary, E-mail: szaszp@inf.u-szeged.hu

Szépkúti, István: H-1149 Budapest, Bíbor u. 18. III/2., Hungary, E-mail: szepkuti@inf.u-szeged.hu

Szkaliczki, Tibor: Computer and Automation Research Institute, 1111 Budapest, Lágymányosi u. 11., E-mail: sztibor@ilab.sztaki.hu

Tanács, Attila: Department of Applied Informatics József Attila University, H–6701 Szeged P.O.Box 652, Hungary, E-mail: tanacs@inf.u-szeged.hu

Tarnay, Katalin: KFKI MSZKI Mérés és Számítástechnikai Kutatóintézet, E-mail: h294tar@ella.hu

Tchanin, Alexandr: Technical University of Budapest Tel : (+36 1) 2095400/ 276, E-mail: alex@proto.sztaki.hu

Tóth, László: Institute of Informatics, József Attila University, H-6701 Szeged P.O. Box 652, Hungary, E-mail: tothl@inf.u-szeged.hu

Tömösy, Gyula: Department of Measurement and Information Systems, Technical University of Budapest, H–1111 Budapest, Műegyetem rkp. 9. Bldg. R,I.113, Hungary Phone +36–1–463 2057, E-mail: tomosy@mmt.bme.hu

Tőke, Pál: Eötvös Loránd University, H-1088 Budapest, Múzeum krt. 6-8, E-mail: toke@heueka.inf.elte.hu

Velickovic, Vesna: Faculty of Philosophy, Department of Mathematics University of Niš, Ćirila i Metodija 2 18000 Niš, Yugoslavia, E-mail: vvesna@kalca.junis.ni.ac.yu

Vincze, Nándor: Juhász Gyula Teachers Training College, Department of Computer Science, Szeged, E-mail: vincze@kurrah.cab.jgytf.u-szeged.hu

Volcansek, Bojan: E-mail: boki@o2edn.edn.ni.ac.yu

Werner, Zsolt: E-mail: werner@ttt-atm.ttt.bme.hu