SCIENTIFIC RESEARCH AT THE INSTITUTE OF INFORMATICS OF THE UNIVERSITY OF SZEGED 2000 – 2003

Systematic education in computer science was launched within the Mathematical Institute at the end of the 1950's by László Kalmár. The Institute of Informatics was founded as an independent unit in 1990. It consists of five departments and a research group:

- Department of Applied Informatics, head of the department: Dr. Balázs Imreh
- Department of Computer Algorithms and Artificial Intelligence, head of the department: Dr. János Csirik
- Department of Foundations of Computer Science, head of the department: Dr. Zoltán Ésik
- Department of Image Processing and Computer Graphics, head of the department: Dr. Attila Kuba
- Department of Software Engineering, head of the department: Dr. Tibor Gyimóthy
- Research Group on Artificial Intelligence of the Hungarian Academy of Sciences, head of the research group: Dr. János Csirik

The main activity of the institute is the education of modern informatics and computer science knowledge. The institute offers B.S. and M.S. degrees. The curricula of the Departments of Informatics consist mainly of mandatory courses. The curricula have already been adjusted to the standards of leading universities, embracing most of the topics dealing with modern informatics and computer science program, e.g. on programming languages, compilers, operating systems, databases, networks, architectures, computer graphics, etc. There are many optional special courses available on the most up-to-date topics in computer science and informatics, e.g. on parallel and distributed computing, expert systems, image processing, advanced operating systems, etc. We offer a number of basic mandatory courses, which place the emphasis on the theoretical aspects of computer science.

The Institute of Informatics offers B.S. and M.S. degrees in the following subjects:

- Computer Science (B.S.)
  - It is a 3-years educational program.
  - The aim of this program is to instruct informatics experts who have up-to-date and high level knowledge in informatics and the related foundations of mathematics.
- Computer Science (M.S.)
  - This educational program takes 5 years.
  - The aim of this program is to educate informatics experts who have up-to-date and high level computer science knowledge which is based on deep foundations of mathematics and computer science.
- Computer Science/Economics (M.S.)
  - This program is 5 years long.
  - The aim of this program is to teach informatics experts who have up-to-date and high level knowledge in informatics and economics based on deep foundations of mathematics.
- Computer Engineering (M.S.)
  - This program is 5 years long.
  - The aim of this program is to teach informatics experts who have up-to-date and high level informatics and technology knowledge based on deep foundations of mathematics.
- Education in Informatics
  - It is a 5-years program.
  - The aim of this education is to instruct the teachers for informatics who have up-to-date knowledge in informatics and education and on the foundations of mathematics.

The main courses on the Bachelor of sciences level are:

Computer architecture, Foundations of programming, Programming, Operations research, Approximate and scientific computing, Algorithms and data structures, Operation systems, Formal languages and their syntactic analysis, Programming languages, Computer networks, Databases, Computer graphics, Complexity theory, System development, Artificial intelligence, Calculus, Discrete mathematics, Linear algebra, Foundation of probability theory, and Mathematical statistics.

The main courses on the Master of sciences level are:

Advanced programming, Elementary informatics, Artificial intelligence, Formal languages and syntactic parsing, Complexity theory, Logic in computer science, Web content development, Machine learning, Medical picture processing, Combinatorics, Game theory, Geometry of fractals, Digital technology, Coding theory, Telecommunication networks, Informatics in economy, Microeconomy, Macroeconomy, Introduction to mathematical economy, and Statistics.

The optional courses are:

Introduction to Informatics, Semantics of programming languages, Scientific and symbolic calculations, Automata and formal logic, Termrewriting systems, DNS computation, Fuzzy theory, Packing and scheduling, Machine learning methods, Natural language processing, Symbolic knowledge representation, Machine learning with a statistical approach, Datamining, Statistic methods of pattern recognition, String processing systems, Multimedia, Database systems, Parallel and distributed computing, Network operating systems, Programming methodology, Computational graphic, Geographic information systems, Computational production control, and Computational statistics.

A more detailed description of the educational programs can be found at the web site of

http://www.inf.u-szeged.hu/oktatas/starten.xml

In addition to the above programs, a doctoral program in Computer Science is available since 1993. The aim of this program is to support postgraduate computer science studies, leading to the degree of Ph.D. in computer science, with an emphasis on theoretical aspects. The Program is part of the Doctoral School in Mathematics and Computer Science of the Faculty of Science of the University of Szeged. It is composed of three subprograms: Theoretical Computer Science, Operations Research and Combinatorial Optimization, and Applications of Computer Science. The possible research topics include mostly those parts of computer science and related areas, which are being investigated at the Institute of Informatics.

The Institute provides also introductory courses in informatics, programming, data processing, and networks. These courses are available for all the students of the university (i.e. not only for computer science students). The programs and research tasks are supported by the scientific and educational cooperations (e.g. CEEPUS, SOCRATES / ERASMUS) with over 20 higher education institutes from Europe, Japan, and the United States:

Boston University, USA,
Columbia University, USA
Dresden University of Technology, Germany,
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany,
Kyoto Sangyo University, Japan,
LaBRI, University of Bordeaux, France
LIAFA, University of Paris, France
Stevens Institute of Technology Hoboken, USA,
Technische Universität Graz, Austria,
Technische Universität Ilmenau, Germany,
Technische Universität Wien, Austria
Universidad de Almería, Spain,
Universität Bern, Switzerland,
Universität Hamburg, Germany,
Universität Karlsruhe, Germany,
Universität Stuttgart, Germany,
University of Aalborg
University of Nis, Yugoslavia,
University of Novi Sad, Yugoslavia,
University of Rome, Italy, and
University of Turku, Finland.

The main research fields investigated in our institute are:

Software Engineering: Static and dynamic analysis of software systems, Optimizing the GCC compiler, parallel and distributed computing.

Staff members working on these fields are: Dr. Zoltán Alexin, Dr. Tibor Gyimóthy, and Dr. Lajos Schrettner.

Artificial Intelligence: string matching, pattern recognition, neural nets, genetic algorithms, medical image processing, machine learning methods and their applications, speech recognition.

Staff members working on these fields are: Dr. Zoltán Alexin, Dr. János Csirik, Dr. Tibor Gyimóthy, Dr. Csanád Imreh, András Kocsor, Dr. Gabriella Kókai, Kornél Kovács, and László Tóth. .

Image Processing and Computer Graphics: automatic map interpretation, medical image processing, analysis and registration, picture processing, and thinning algorithms, GIS (Geographic Information Systems).

Staff members working on these fields are: Dr. Zoltán Kató, Dr. Endre Katona, Dr. Attila Kuba, Dr. Eörs Máté, Dr. László Nyúl, and Dr. Kálmán Palágyi.

Operations Research: combinatorial optimization, process network synthesis, nonlinear and global optimization, interval mathematics, fuzzy systems.

Staff members working on these fields are: Dr. Tibor Csendes, Dr. József Dombi, Dr. Balázs Imreh, Dr. Csanád Imreh, Dr. Zoltán Kovács, and Dr. András Pluhár.

Theoretical Computer Science: algebraic theory of automata, automata and formal logic, formal languages and formal power series, tree automata and tree transducers, term rewriting systems, equational theory of fixed points, iteration theories, formal semantics and process algebras, algebraic theory of sequential circuits, attributed grammars, complexity theory.

Staff members working on these fields are: Dr. Zoltán Ésik, Dr. Zoltán Fülöp, Dr. Ferenc Gécseg, Dr. Balázs Imreh, Dr. Sándor Vágvölgyi.

Networking: Migration from IPv4 to IPv6.

Staff members working on these fields are Dr. Mihály Bohus and Dr. Lajos Schrettner.

A more detailed description of the research activities of the Institute of Informatics can be found at the web site of

http://www.inf.u-szeged.hu/kutatas/starten.xml

The Institute of Informatics has a library which holds about 5000 Hungarian and English language volumes and subscribes over 200 scientific journals. A scientific journal, Acta Cybernetica has been published since 1969 by the institute in English. The journal is available in about 150 university departments worldwide.

More detailed information on the Institute of Informatics can be found at its web site

http://www.inf.u-szeged.hu/starten.xml

APPLICATION OF MACHINE LEARNING ALGORITHMS AND NATURAL LANGUAGE PROCESSING

Zoltán Alexin, János Csirik, Tamás Gergely, Tibor Gyimóthy, Ferenc Havasi, András Hócza, and Miklós Kálmán

The application of ML (Machine Learning) to practical problems is an emerging field of research and development. In many cases only machine learning can provide feasible approaches to algorithms, testing, and validation. Researchers at the department are studying a number of areas related to software engineering like the compression of large XML and code files, and the processing of natural language texts.

I. Compacting XML Documents

XML is a widely used format for storing information. The biggest drawback of XML documents is that their size is rather large compared to the information they store. XML documents may contain redundant attributes which can be calculated from others. These redundant attributes can be deleted from the original XML document if the calculation rules can be stored somehow. In an Attribute Grammar environment there is an analogue description for these rules: *semantic functions*. In order to use this technique in an XML environment a new metalanguage called SRML[8] (Semantic Rule Meta Language) has been defined. Researchers at the department developed a software method which makes this SRML approach suitable for compacting large XML documents.

II. Compression of machine executable codes

The aim of the research was the development of an effective ARM code compressor.

Compressors usually consist of two parts: a model and a coder. The model part provides statistical information about the code for the coder while the coder part is responsible for the physical coding. The model part of our compressor is a decision tree and the coder part is an arithmetic coder.

The building of the decision tree is the usual, entropy-based greedy tree-building algorithm (like ID3) adapted to our requirements. It creates a full-size tree for a given training data set S.

The tree that is built is then used as a model for the arithmetic coder. During the tree building process the size of the code and the tree are minimized together.

This compression algorithm is best for situations where a large amount of code is stored in read-only form and the storage space is more important than the decompression time.

This algorithm was implemented and used in the compressed file system of an ARM-based embedded machine. It resulted in an additional 10% size reduction: the gzipped images were 55% of the original 25MB images, while those compressed with ARMLIB were 45%.

III. Natural Language Processing

Investigating characteristic features of natural language by means of information technology is a tradition at the Department of Informatics. The history of it goes back to László Kalmár's work in the seventies. Since then the work has been continuously going on at varying intensity. The intensity of research into this area substantially increased in 1998 when the department, as a participant of the ESPRIT LTR20237 "ILP2" project, began to deal with the application of machine learning algorithms to NLP (Natural Language Processing).

The Department of Informatics initiated a Hungarian NLP consortium in 1999 whose members are the following institutions: Research Institute for Linguistics at the Hungarian Academy of Sciences, MorphoLogic Ltd Budapest, and the Department of Informatics at the University of Szeged. The department is a founding member of the international TEI (Text Encoding Initiative) Consortium (`http://www.tei-c.org`) as well.

The Hungarian NLP consortium successfully applied for R&D grants from the Hungarian Ministry of Education. Three major projects (OM IKTA 27/2000, NKFP 2/017/2001, OM IKTA 37/2002) won substantial support which enabled the manually annotated corpus works to be developed and also provided financial support for research, e.g. the application of machine learning algorithms to NLP.

In 2003 the Department of Informatics organized the First Hungarian Conference on Computational Linguistics in Szeged. About hundred participants from all over Hungary came and presented an overview of their work [2]. A special issue of the international journal Acta Cybernetica will be published in 2004 containing the best papers of the conference.

IV. The Szeged Corpus — A Manually Annotated Hungarian Learning Corpus

Development work of the manually annotated Hungarian corpus began in 2000. At that time there was no sufficiently large morpho-syntactically annotated (disambiguated) corpus for Hungarian. Such a database is a must for further natural language and machine learning processing. In order to create such a training corpus, consortium partners began a 2-year project (OM IKTA 27/2000). The 1 million-word corpus was ready in 2002. Since then it can be freely obtained for research and education purposes after registration [1].

The corpus is available in XML and was encoded using the TEIXLITE[1] DTD scheme. Texts were collected from five different areas of recent Hungarian written sources (fiction, newspapers, legal texts, computer-oriented texts, teenage compositions). Most of the texts are from 1999. Each word in the corpus is tagged by its morpho-syntactic labels, and then disambiguated. The HLT group built a software environment that supported the annotation work [4].

A newer version (2.0) of the corpus has recently been completed. This version not only contains 20% more text words but every hierarchic NP (noun phrase) annotation as well [6]. Since NPs are key elements of the sentence syntax, a big corpus like this is an authentic learning database for determining NP syntax rules. For more information about this corpus and further work: `http://www.inf.u-szeged.hu/hlt`.

In the framework of the OM IKTA 37/2002 project the consortium is working on the Hungarian treebank, that is a syntactically analyzed and annotated corpus. In the treebank all ADVPs (adverbial phrases), ADJPs (adjectival phrases), PPs (postnoun phrases), CLs (clausal phrases), and VPs (verb phrases) are annotated. This work should be completed in 2004. Using this corpus as a training and validation database the consortium hopes to develop syntax rules and an effective syntactic parser for Hungarian sentences.

V. Parsing Natural Language Texts

Researchers are studying the application of formal techniques at different stages of NLP. Besides the preprocessing of texts, three particular tasks are mentioned below where the application of formal approaches seems to be promising, as reported in many publications.

V.1. Part-of-speech Tagging

The task of a POS tagger is, for a given text, to provide for each text word its contextually disambiguated part-of-speech *tag*. In the first experiments in 1999, the annotated Hungarian TELRI corpus from the MULTEXT-East project was used. The TELRI corpus contains about 100,000 tokens (words and punctuations) which is relatively small compared to the Wall Street Journal corpus for English, say, with its 3 million words. In a comparison study [4] the efficiency of 5 different learning algorithms was tested in learning disambiguation rules for Hungarian. Each algorithm produced rules like these and, by applying these rules in a working tagger, a 96-97% per-word accuracy could be attained. The work was done in cooperation with researchers from the German GMD institute in Bonn.

In [4], authors presented a preliminary study designed to illustrate the capabilities and limitations of current ILP and non-ILP algorithms on the Hungarian POS-tagging task. The popular C4.5 and Progol systems as propositional and ILP representatives were selected, adding experiments with methods AGLEARN [5], a C4.5 preprocessor based on *attribute grammars*, and two ILP approaches PHM and RIBL [12, 13].

After the manually annotated corpus was ready for use the consortium studied several different learning algorithms for making a real-world POS-tagger program for Hungarian. After several attempts at using first-order (ILP) learning methods, a cascade of a TnT

(`http://www.coli.uni-sb.de/ thorsten/tnt`)

and a Brill tagger

(`http://www.cs.jhu.edu/ brill`)

currently gives the best results trained on a larger part of the 1.2 million-word corpus.

V.2. NP-Chunking, Shallow Parsing

Similar to many other languages, Hungarian also relies heavily on the use and interrelation of elementary word structures (syntagmas). NPs (Noun Phrases) are the most distinctive units of Hungarian sentences, hence shallow parsing primarily focuses on NP recognition. NPs are linguistically well defined. (See Research Report on Natural Language Processing.)

Researchers are currently studying two different methods for learning NP syntax rules. Namely, using a specially designed regular expression learning algorithm [10], and a tree compression method. The latter one is based on the frequencies of subtrees, and is strongly connected to the research being done on compacting XML files and byte codes. Both ML methods are currently under investigation. To test syntax rules a bottom-up parser engine is being developed as well. Since the NP grammar is ambiguous, built-in heuristics are employed to choose the most appropriate parsing of input sentences.

An NP is an identified group in a morpho-syntactically annotated sentence which contains one or more words or punctuation marks. According to the linguistic definition for Hungarian, a group within a sentence can be an NP if

- it has a noun head (i.e. the last word in the

[1] http://www.tei-c.org

group is a noun),

- it is contiguous (i.e. the words and/or punctuation marks are located next to each other),
- the size of the group is maximal (i.e. it includes all determiners, adjuncts, modifiers and/or specifiers belonging to the noun head).

For example, in the Hungarian sentence *'Ez azt jelenti, hogy a határőrök minden személy úti okmányát egyenként megvizsgálják'* (This means that the customs officers check the passport of each person one at a time) the following NPs can be identified: *Ez* (this), *azt* (that), *a határőrök* (the customs officers), and *minden személy úti okmányát* (the passport of each person).

Researchers are working on a rule-based NP recognizer program that unites syntax rules elaborated by linguist experts with the rules learned by machine learning algorithms. To validate the software the NP annotated Hungarian corpus is used.

V.3. Information Extraction

The aim of the IE (Information Extraction) is to provide a structured, queriable dataset from the content of natural language texts. This does not require a program to fully understand the text. One approach to IE is to set up templates with slots to be filled with information – so-called *semantic frames* –, then try to match documents to these templates. Machine learning algorithms have been successfully used in learning semantic frames from training data. Researchers at the department created a learning dataset with 2000 short business news items as a basis for further investigation. Studying the applicability of ML in learning semantic frames is also being investigated[9].

With the exponential growth of online available information, Information Extraction (IE) has become a focus of interest. IE is located somewhere between Information Retrieval (IR) and Natural Language Understanding (NLU): it aims to provide a structured, queriable dataset from the contents of natural language texts without the requirement of fully understanding it. The resulting dataset contains only relevant information instead of entire sentences. This means that IE is a shallow form of understanding, which could be used for processing certain types of documents. An IE system is a given set of documents and template of slots to be filled with information (so-called semantic frames) and the system which returns a filled template with the extracted information from each document. Synthesizing simpler natural language sentences from the extracted information, one can obtain a more concentrated version of the original text. Both IE and text compression by IE are, currently, intensively studied areas of NLP.

A set of NLP modules linked together is required for a working IE system, such as segmentation module, morpho-syntactic parser, named entity recognizer, POS-tagger, shallow parser, and semantic annotator. The latter one can add background information to certain NP (or other types of) nodes in the parse trees of input sentences. The IE task can be carried out by defining the syntactic and semantic attributes of slots occurring in the same template. Templates are widely known as *semantic frames*. The final step in IE is the semantic frame matching by an algorithm that selects one semantic frame at a time and tries to match it with the parsed (preprocessed) sentence. The best-fit frames are selected then ordered, say, according to the amount of information have been extracted.

The Department of Informatics together with its consortium partners developed a tool-chain of NLP modules necessary for a prototype of an IE system and an IE prototype system. This prototype can extract information from short business news text concerning some events in business life, such as acquisitions, balance reports, mid-term reports, contracts, or fusion. The development work of the NewsPro system was supported by the grant OM NKFP 2/017/2001.

VI. Research Connections in Education

The above research relates to the following fields in Computer Science: Artificial Intelligence, Machine Learning, Inductive Logic Programming, Knowledge Discovery, Computational Linguistics, Information Extraction, Formal Languages, Theory of Compiling and Parsing and Logic Programming.

Unsolved problems are being studied by PhD students and investigated in students' scientific projects, in masters and bachelor theses. The group of researchers interested in different areas of NLP is called the Human Language Technology Group. Visit the homepage of the group at URL: `http://www.inf.u-szeged.hu/hlt`. The web site contains information about the members, publications, projects and downloadables provided by the group.

Acknowledgement

The above-mentioned research were partially supported by the following grants: ESPRIT LTR20237 "ILP2", ILPNet2 EU INCO 9771002, OMFB IKTA 8/99, OM IKTA 27/2000, NKFP 2/017, OM IKTA 37/2002.

REFERENCES

[1] Alexin, Z., Csirik, J., Gyimóthy, T., Bibok K., Hatvani, Cs., Prószéky, G., Tihanyi, L.: Manually Annotated Hungarian Corpus in Proceedings of the Research Note Sessions of

the 10th Conference of the European Chapter of the Association for Computational Linguistics EACL'03, Budapest, Hungary 15-17 April, pp. 53–56 (2003)

[2] Alexin, Z., Csendes, D. (editors): Proceedings of the MSZNY 2003 (Hungarian Computational Linguistics) Conference, Szeged, Hungary, 10-11 December, (2003)

[3] Alexin, Z., Gyimóthy, T., Váradi, T., Oravecz, Cs.: A rule-based tagger development framework, a late-breaking paper, in Proc. of the First Workshop on Learning Language in Logic LLL99 Eds: Sašo Džeroski, Bled, Sloven, pp. 1–7, (1999)

[4] Alexin, Z., Leipold, P., Csirik, J., Bibok, K., Gyimóthy, T.: A rule-based tagger development framework in Proc. of the Third Workshop on Learning Language in Logic LLL'2001 Eds: Luboš Popelíský, Strasbourg, France, pp. 1–10, (2001)

[5] Alexin, Z., Zvada, Sz., and Gyimóthy, T.: Application of AGLEARN on Hungarian Part-of-speech Tagging, in Proceedings of the Second Workshop on Attribute Grammars and their Applications, WAGA'99, Amsterdam, The Netherlands, Eds.: D. Parigot and M. Mernik, INRIA Rocquencourt, pp. 133–152, (1999) *http://www-rocq.inria.fr/oscar/www/fnc2/WAGA99/proceedings/alexin/alexin.ps*, *http://www-rocq.inria.fr/oscar/www/fnc2/WAGA99/proceedings/alexin/alexin.pdf*

[6] Csendes, D., Hatvani, Cs., Alexin, Z., Csirik, J., Gyimóthy, T., Prószéky, G., Váradi, T.: Manually Annotated Hungarian Natural Language Corpus: the Szeged Corpus, in Proceedings of the MSZNY 2003 (Hungarian Computational Linguistics) Conference, Szeged, Hungary, 10-11 December, pp. 238–247 (2003)

[7] Gyimóthy, T. and Horváth, T.: Learning Semantic Functions of Attribute Grammars *1997. Nordic Journal of Computing Vol 4. pp. 287–302*

[8] Havasi, F.: XML Semantics Extension *2002. Acta Cybernetica Vol 15. pp. 509-528*

[9] Hócza, A., Alexin, Z., Csendes, D., Csirik, J., Gyimóthy, T.: Application of ILP methods in different natural language processing phases for information extraction from Hungarian texts, in Proceedings of the Kalmár Workshop on Logic and Computer Science, Szeged, Hungary, 1-2 October, pp. 107–116, (2003)

[10] Hócza, A., Iván, Sz.: Learning and recognizing noun phrases, in Proceedings of the MSZNY 2003 (Hungarian Computational Linguistics) Conference, Szeged, Hungary, 10-11 December, pp. 72–79, (2003)

[11] Horváth, T., Alexin, Z., and Gyimóthy, T, Wrobel, S.: Application of Different Learning Methods to Hungarian Part-of-Speech Tagging, in Proc. of the 9th International Workshop on Inductive Logic Programming, Eds: Sašo Džeroski and Peter Flach, LNAI, **1634**, pp. 128–139, Springer Verlag, ISBN: 3-540-66109-3, (1999)

[12] Horváth, T. and Turán, Gy.: Learning logic programs with structured background knowledge in *Artificial Intelligence*, **128**, pp. 31–97, (2001)

[13] Horváth, T., Wrobel, S., and Bohnebeck, U.: Relational instance-based learning with lists and terms in *Machine Learning*, **43**, pp. 53–80, (2001)

[14] Kuba, A., Bakota, T., Hócza, A., Oravecz, Cs.: Comparing different POS-tagging techniques for Hungarian, in Proceedings of the MSZNY 2003 (Hungarian Computational Linguistics) Conference, Szeged, Hungary, 10-11 December, pp. 16–24, (2003)

OPEN-SOURCE CODE OPTIMIZATION

Árpád Beszédes, Tamás Gergely, Tibor Gyimóthy, Ferenc Havasi, Ákos Kiss, and Gábor Lóki

I. INTRODUCTION

Reducing the size of program code is important from several aspects, such as reducing the network traffic and the ability to produce software for embedded systems that require little memory space and are energy-efficient. Research in this field can be divided as *code compaction* methods which reduce the program code by transformations giving equivalently executable code, and *code compression* where the code is stored in a compressed format and decompressed on demand. With both approaches we developed practically usable methods in cooperation with Nokia Mobile Phones, Helsinki in a R&D project during 2002–2003. This work resulted three U.S. patent applications and an open-source compression framework project maintained by the department. We also prepared a survey paper about code-size reduction methods which appeared in the ACM Surveys journal [1].

The size of the program code in its executable binary format highly depends on the compiler's ability to produce compact code. Compilers are generally able to optimize for code speed or code size. However, performance is more investigated and little effort is being put on optimizing for code size. We have significant achievements in improving the open source compiler, GCC in this respect.

Two related courses to open source software are taught at the university by our department. The first, Linux Kernel Course, is to introduce the students of the basics of Linux Kernel with active work. It covers introductory and advanced topics about the internals of this operating system. The other course is about Open-Source Software Development, where the students are introduced to the basics of open-source software development with active work. Topics include an overview of open source software development processes with the details of build and configuration management, bug tracking, security issues and open source licencing policies. Important open source projects are also introduced such as GCC, the Linux Kernel, Mozilla, etc.

II. CODE COMPRESSION

We developed a code compression method for the ARM instruction set, which is more efficient than general compressors [5]. It is a specialized compressor composed of two parts, a model and a coder. The model part provides probability information about the code for the coder, while the coder part is responsible for the actual encoding of the sequence. The model part of our compressor is a decision tree with an enhanced pruning method, while as the coder we use an arithmetic one. This compression algorithm performs best in situations where a big amount of code is stored in a read-only way (because the complex training phase involves long compression time) and the storage space is more important than the decompression time. This is because this model-based compressor specific to ARM code produces compressed data with significantly better compression ratios, but the decompression time is slightly worse compared to a general (e. g. dictionary based) compressor. This is the case with the JFFS2 file system as well, as overviewed below.

The compressor is also part of an open source compression framework, which is intended to be used in JFFS2 (Journalling Flash File System, version 2), a file system designed for use on flash devices in embedded systems. This framework, called BBC, is able to extend JFFS2 with the capability of adding new block-based compressors to the file system, including the decision tree-based one overviewed above. The goal of this research was to improve the performance of JFFS2. JFFS2 already has a compression feature, it uses ZLIB, a common compression library. In BBC ZLIB is replaced with a compressor framework, which is able to use an arbitrary set of compressors. The most appropriate compressor can be chosen based on the requirements: the compression ratio, the read or write speed. For example, if compression ratio is important it will try all compressors and choose the one which produces the smallest result. Using this JFFS2 extension we achieved relevant speed and/or size reductions. The implementation will be accessible at our website [9] as of March, 2004.

III. CODE COMPACTION

With this research a code compaction tool was developed that is able to compact binary executable codes for the ARM/Thumb architecture [3, 4]. This tool reduces the memory footprint of binary executables by appropriate code transformations that produce functionally equivalent but shorter code. Code compaction is a transformation that modifies program code so that it will contain fewer instructions but will run as usual. Compaction differs from compression, where some kind of decompression is needed before execution. The compaction techniques used are the same as used by compilers when optimizing for size, but since we process the whole executable inter-module optimizations are possible. The basis for all transformations is a precise Control Flow Graph, which is built using sophisticated algorithms and heuristics. With the help of this graph various program transformations are made, which include *dead code elimination*, *unreachable code elimination*, *common sequence abstraction*, etc.

IV. COMPILER MEASUREMENT AND OPTIMIZATION

GCC (GNU Compiler Collection), the open source compiler can be instructed to optimize the generated code for its size. Since the majority of the compiler's developers are interested in the performance of the generated code, this objective is often neglected. Several benchmarks exist that measure the performance of the generated code on a daily basis, however benchmarks for code size did not exist previously. A benchmark (called CSiBE, GCC Code Size Benchmark) has been developed at the department which regularly measures the size of the code generated by GCC [6]. The benchmark consists of a test bed of several typical C applications, a database which stores daily results and an easy-to-use web interface with sophisticated query mechanisms. GCC source code is checked out daily from the central source code repository, the compiler is built and measurements are performed on the test bed. The results are stored in the database (they are available starting from May 2003), which is accessible via the CSiBE website using several kinds of queries. Code size and compilation time data is available via raw data tables or using appropriate diagrams generated on demand. We continuously maintain the benchmark, look for irregular changes in the results and act if required. Until now we got only positive feedback from the GCC community, it is now considered as the standard GCC benchmark for code size. Thanks to the existence of this benchmark, the compiler was improved several times to generate smaller code, either by reverting some fixes with side effects or by using it for fine tuning some algorithms. The benchmark is also accessible from the GCC benchmarks homepage [8]. We presented our work related to the measurement of the code size generated by GCC in a talk at the first international GCC summit in 2003 [2].

We run another website on which problems related to the GCC port for the ARM architecture are collected [7]. We identified several existing problems and possibilities for further enhancement of the compiler and made them accessible to the community. Several enhancements to the optimization algorithms in GCC were developed by the department, which have officially been accepted by the community and are now part of GCC.

REFERENCES

[1] Árpád Beszédes, Rudolf Ferenc, Tibor Gyimóthy, André Dolenc, and Konsta Karsisto. Survey of code-size reduction methods. *ACM Computing Surveys*, 35(3):223–267, September 2003.

[2] Árpád Beszédes, Tamás Gergely, Tibor Gyimóthy, Gábor Lóki, and László Vidács. Optimizing for space : Measurements and possibilities for improvement. In *Proceedings of the 2003 GCC Developers' Summit*, pages 7–20, May 2003.

[3] Attila Bicsák, Ákos Kiss, Rudolf Ferenc, and Tibor Gyimóthy. A Method and a System for Constructing Control Flows of Binary Executable Programs at Post-Link Time. Technical report, University of Szeged, August 2003.

[4] Attila Bicsák, Ákos Kiss, Gábor Lehotai, and Rudolf Ferenc. Method and Device for Abstracting Instruction Sequences With Tail Merging. Technical report, University of Szeged, September 2003.

[5] Tamás Gergely, Ferenc Havasi, and Árpád Beszédes. A Model Based Code Compression. Technical report, University of Szeged, October 2003.

[6] Homepage of CSiBE. `http://sed.inf.u-szeged.hu/CSiBE`.

[7] GCC-ARM Improvement Project homepage. `http://sed.inf.u-szeged.hu/gcc-arm`.

[8] GCC benchmarks homepage. `http://gcc.gnu.org/benchmarks`.

[9] Homepage of BBC. `http://www.inf.u-szeged.hu/jffs2`.

STATIC AND DYNAMIC PROGRAM SLICING

Árpád Beszédes, Tamás Gergely, Tibor Gyimóthy, Ákos Kiss, and Gyöngyi Szilágyi

I. TOPICS AND PEOPLE

Program slicing is a powerful means for the debugging, maintenance, reverse engineering and testing of programs. A slice consists of all statements and predicates that might affect a set of variables at a certain program point. A slice may be an executable program or a subset of the program code. Static slicing methods compute those statements which influence the value of a variable occurrence for all possible program inputs, while dynamic slicing methods take only one specific input into account.

The program slicing activities of the department involve the improvement of existing slicing methods and the invention of new solutions. Several researchers and highly skilled programmers take part in the research and in addition, PhD and MSc students are also involved in a number of topics.

II. SLICING OF C PROGRAMS

An efficient algorithm has been elaborated for dynamic slicing, whose main advantage over prior methods is that it can be applied to real-size C programs because its memory requirements are proportional to the number of different memory locations used by the program, rather than the number of executed steps, which may be unbounded. The algorithm presented is a forward global method for computing backward dynamic slices of C programs. In parallel to the program execution the algorithm determines the dynamic slices for any program instruction [2]. Research has also been done on the handling of unstructured statements and pointers [3, 4] in the algorithm.

In many cases static slices are overly conservative and therefore too large to supply useful information. Dynamic slicing methods can produce more precise results, but only for one test case. Therefore the concept of union slices has been introduced [1]. Using a combination of static and union slices the size of program parts that need to be investigated can be reduced significantly.

III. SLICING OF BINARY EXECUTABLES

Although the slicing of programs written in high-level languages has been widely studied in the literature, very little work has been done on the slicing of binary executable programs. At machine level central notions such as control structures and variables are missing, therefore the analysis of machine code has to work with fully unstructured control flow, registers and memory locations. In [10, 11] a method is presented for the interprocedural static slicing of binary executables. The papers describe the problems arising during control flow, control dependence and data dependence analysis and provide solutions. Additionally, a method is introduced to extend the conservative analysis of registers to the local stack image of procedures thus making possible to take local registers residing on stack into account. A slicing tool for ARM binaries is implemented and results on measurements of static slice sizes of binary executables are given.

An extended version of this paper, which discusses how static slices of binaries can be made more precise using dynamically gathered information has been invited to a special issue of the Software Quality Journal [9].

IV. SLICING OF (CONSTRAINT) LOGIC PROGRAMS

Slicing is a program analysis technique originally developed for imperative languages, but the data flow in logic programs is not explicit, and for this reason the concept of a slice and the slicing techniques of imperative languages are not directly applicable. Moreover, implicit data flow makes the understanding of program behavior rather difficult. Thus program analysis tools explaining data flow to the user are of great practical importance. Papers [8, 5] extends the scope and optimality of previous algorithmic debugging techniques of Prolog programs using slicing techniques. They provide dynamic slicing algorithms augmenting the data flow analysis with control flow dependence to help one locate the connected components of a program and the source of a bug included in the program. A tool has been developed for debugging Prolog programs which also handles the specific programming techniques (cut, if-then, or).

Constraint logic programming is a fusion of two declarative paradigms: constraint solving and logic programming. The framework extends classical logic programming by removing the restriction on programming within the Herbrand universe alone; unification is replaced by the more general notion of constraint satisfaction. In [6, 7] declarative notions of a slice suitable for CLP are formulated. (The problem of finding minimal slices may be undecidable in general, since satisfiability may be undecidable.) The presented slice definition provides a basis for defining slicing techniques based on variable sharing. The techniques are further extended by using groundness information. A prototype slicing tool for CLP programs written in SICStus Prolog is also presented.

REFERENCES

[1] Árpád Beszédes, Csaba Faragó, Zsolt Mihály Szabó, János Csirik, and Tibor Gyimóthy. Union slices for program maintenance. In *Proceedings of the IEEE International Conference on Software Maintenance (ICSM 2002)*, pages 12–21, October 2002.

[2] Árpád Beszédes, Tamás Gergely, Zsolt Mihály Szabó, János Csirik, and Tibor Gyimóthy. Dynamic slicing method for maintenance of large C programs. In *Proceedings of the Fifth IEEE European Conference on Software Maintenance and Reengineering (CSMR 2001)*, pages 105–113, March 2001.

[3] Csaba Faragó and Tamás Gergely. Handling the unstructured statements in the forward dynamic slice algorithm. In *Proceedings of the Seventh Symposium on Programming Languages and Software Tools (SPLST 2001)*, pages 16–27, June 2001.

[4] Csaba Faragó and Tamás Gergely. Handling pointers and unstructured statements in the forward computed dynamic slice algorithm. *Acta Cybernetica*, 15:489–508, 2002.

[5] Gyöngyi Szilágyi, László Harmath, and Tibor Gyimóthy. Debug slicing of logic programs. *Acta Cybernetica*, 15(2):257–278, 2001.

[6] Gyöngyi Szilágyi, Jan Maluszynski, and Tibor Gyimóthy. Slicing of constraint logic programs. In *Proceedings of the Fourth International Workshop on Automated Debugging (AADEBUG 2000)*, pages 176–187, 2000.

[7] Gyöngyi Szilágyi, Jan Maluszynski, and Tibor Gyimóthy. Static and dynamic slicing of constraint logic programs. *Journal of Automated Software Engineering*, 9(1):41–65, January 2002.

[8] László Harmath, Gyöngyi Szilágyi, and Tibor Gyimóthy. Debug slicing of logic programs. In *Volume of Extended Abstracts of the Second Conference of PhD Students in Computer Science (CSCS 2000)*, pages 43–44, July 2000.

[9] Ákos Kiss, Judit Jász, and Tibor Gyimóthy. Using dynamic information in the interprocedural static slicing of binary executables. *Software Quality Journal*, (submitted).

[10] Ákos Kiss, Judit Jász, and Gábor Lehotai. Static slicing of binary executables. In *Volume of Extended Abstracts of the Third Conference of PhD Students in Computer Science (CSCS 2002)*, page 60, July 2002.

[11] Ákos Kiss, Judit Jász, Gábor Lehotai, and Tibor Gyimóthy. Interprocedural static slicing of binary executables. In *Proceedings of the Third IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2003)*, pages 118–127, September 2003.

NETWORKS AND DISTRIBUTED COMPUTING

Vilmos Bilicki and Lajos Schrettner

I. SIP Compression

The wired line network is well studied and what is the most important it is widely used for along time. Most of its protocols passed the test of time. There are many similar tasks in mobile and wired line environment, and we would like to achieve compatible, inter working solutions. So it is a plausible idea to use these protocols in mobile network too. The mobile environment differs from wired environment significantly. The main difference is in the bandwidth, which is going to be smaller with the help of the new generation of mobile networks (GERAN 2 Mbps), but it will still remain significant. An acceptable solution is to compress these protocols.

The goal of our work was to study the SIP protocol and based on this research, to develop and implement efficient compressing algorithms. We implemented a test framework for both dynamic and static compression. The research project was funded by Nokia Hungary. For further details please consult to the article [1].

II. Distributed Computing

Parallel programming languages have two advantages over sequential ones. One is that a parallel program can run faster if suitable hardware is available, the other is that as programs often model real world phenomena, the solution may be expressed more naturally using explicit or implicit parallelism. Despite this potential, parallel programming has not become widespread because there are no general purpose parallel programming models.

A general purpose parallel architecture has to be extendible, an ideal general purpose parallel programming model on the other hand has to hide the architectural details from programmers. If these requirements are satisfied, portable parallel programs can be written, opening the way to widespread use of parallel computing. By using extendible machines, their speed and capacity can be adjusted by the users to their needs.

We proposed the NOP programming model [2], which incorporates a number of novel features. It has a single data structure that makes random memory accesses unnecessary. This radical approach has both advantages and disadvantages, but it is considered a viable alternative method of memory management in the context of parallel processing. The model has both imperative and declarative features, trying to combine the best of both worlds. Its simple imperative transformations can be easily and efficiently implemented, while larger programs can be constructed in a declarative style, benefiting from the substantial body of accumulated knowledge in the area of functional programming languages. By studying the existing programming models, we can observe that random access causes implementation difficulties in distributed and even in shared memory environments. A more regulated, more predictable flow of data would be more satisfactory. Further, the imperative style of programming is preferable if we want performance, declarative style is better at ensuring sound design and program correctness.

The NOP model uses implicit references exclusively. The operations are commands in the imperative sense, they cause some action(s) to be performed. There are sequential, conditional and parallel composition operations. On the other hand, a NOP program is a mutually recursive equation set, in this sense it is declarative. Put into other terms, a functional program (equation set) determines the control structure of the imperative program, which is then executed to map the input into the final result. Using equations, programmers extend the transformation set of the model. These design choices are believed to produce a unique, radical, but satisfying model.

The ultimate goal of the model is of course to allow efficient multiprocessor implementations. A fully functional SM-MIMD NOP executor has been developed and implemented to carry out experiments and gain insight for further research. Research progresses toward developing a full programming environment and investigating a combination of program transformations and dynamic load balancing to arrive at a distributed memory multiprocessor implementation.

REFERENCES

[1] M. Fidrich, V. Bilicki, Z. Sógor, G. Sey. SIP compression. In *Third Conference of PhD Students in Computer Science (ICSM 2002)*

[2] L. Schrettner The NOP Parallel Programming Model. In *Distributed and Parallel Systems, The Kluwer International Series in Engineering and Computer Science, Kluwer, 2000, pp. 127-136*

GLOBAL OPTIMIZATION AND CIRCLE PACKING

Tibor Csendes, Mihály Markót, Péter Gábor Szabó, Boglárka Tóth, and Tamás Vinkó

This summary covers results achieved during the period of 2000-2003 in the fields of developing nonlinear and global optimization procedures, their application in circle packing problems, and in the investigation of their convergence properties. The majority of the studied algorithms are guaranteed reliability methods. This feature — not available for procedures based only on real arithmetic operations — was made possible by the use of interval arithmetic.

The basic problem is usually given in the form of

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

where $f : I\!R^n \to I\!R$ is an (often two times continuously differentiable) real function, and $X \subseteq I\!R^n$ is the set of feasible points. The set X is sometimes a simple n-dimensional interval, while in other cases it is determined by constraints like $g_i(x) \leq 0$ and $h_j(x) = 0$. Global optimization deals with such solutions of problem (1) which are not only locally optimal, but where the related objective function has a globally minimal value.

The above problem class is very general, most of the mathematical problems can be reformulated as such nonlinear optimization problems. This is the reason why we cannot expect to have an algorithm that is capable to solve the whole problem class. Even for simple quadratic problems no polynomial solution algorithm exists. For general nonlinear problems the situation is even worse: such problems are not solvable at the costs of a finite number of function evaluations. The aim of the research is presently to widen the set of solvable problems with new algorithms, and on the other hand, to improve the reliability and the efficiency of the related methods.

The nonlinear and global optimization algorithms have a wide application area, let us just mention the system identification and parameter estimation problems often used in natural sciences. The completed procedures can be downloaded by anonymous ftp from

`ftp.jate.u-szeged.hu`

from the directory of

`/pub/math/optimization`

The clustering global optimization code available from here is downloaded 1-2 times daily, and in two independent comparison studies it proved to be the most efficient ("Of the programs tested, the Derivative-Free Boender-Timmer-Rinnoy Kan Algorithm by Tibor Csendes is the clear winner." is written in `http://www.mat.univie.ac.at/~neum/glopt.html`, and see also the paper of Mongeau et al.[1]).

[1] Mongeau, M., H. Karsenty, V. Rouzé, J.-B. Hiriart-Urruty: Comparison of public-domain software for black-box global optimization. Optimization Methods & Software 13(2000) 203-226

I. INTERVAL METHODS FOR OPTIMIZATION

The largest set of results belong to interval methods for global optimization. These techniques are based on a branch-and-bound algorithm, and the lower and upper bounds for the subproblems are calculated with either the interval arithmetic, or by more sophisticated inclusion functions based on it (see e.g. [7]).

The present investigations aim to improve the efficiency of the guaranteed reliability algorithms, and in the same time enlarge the set of solvable problems. The first such task was to clear the role of multisection, i.e. the technique when not two (as in the case of bisection), but a few more subintervals are produced in each iteration. The theoretical convergence results and the numerical testing was summarized in [2, 10].

In [13] we have investigated with an extensive a posteriory analysis which one-step-look-ahead decision rule would be the best to find out the multisection grade s. The answer is that the rule must produce the maximal possible difference in the inclusion function values for the resulting subintervals: the upper for one of the intervals must be as close to the lower bound of the other as possible.

Also the earlier mentioned multisection technique was improved by the newly found indicator, the so-called *Rejectindex*

$$pf(f^*, X) = \frac{f^* - \underline{F}(X)}{\overline{F}(X) - \underline{F}(X)}$$

first published by L.G. Casado and coworkers [4]. Its application ways to obtain better efficiency for other parts of the B&B method were discussed in [5, 6, 6, 30, 31]. The latter paper is important, since it is the first successful approach since over 27 years to improve the interval selection step of the algorithm.

In the next article [18] the subinterval selection rule was extended for a substantially more general case (when the global minimum value is unknown). The numerical efficiency of the suggested algorithmic change was documented in [17]. The optimality of the new subinterval selection rule was proven in [41]. The above results were generalized for the constrained optimization problems in [29].

Interval branch-and-bound methods can also be applied to solve practical constrained nonlinear optimization problems with tolerances on the decision variables. We have reported our experiences on an engineering problem in [12].

A new interval inclusion function was introduced in [39], and its generalization for multidimensional cases was given in [40].

The foundations of the interval methods and solutions to numerical problems were summarized in the papers [9, 12] (in Hungarian).

II. CIRCLES PACKING IN THE UNIT SQUARE

The circles packing in the unit square problem is an old difficult field of computational geometry. It is also highlighted by the fact, that the ten circles case was

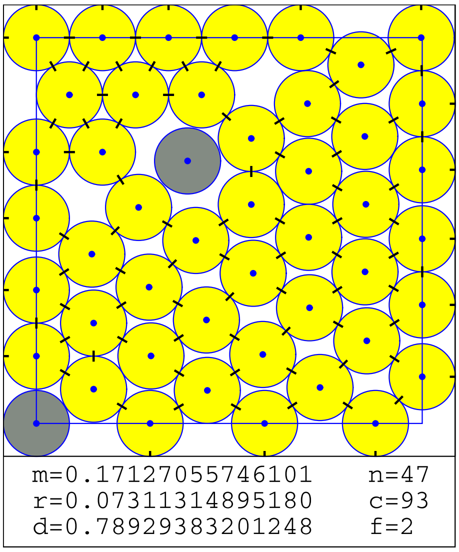

Figure 1. The found improved packing of 47 circles in the unit square. Here r is the common radius of the circles, d is the density of the packing, c stands for the number of connections between circles or between a circle and a side, and f denotes the number of free circles.

first solved in 1990, and that no proven solutions exist above 30 circles (with the exception of $n = 36$). Our first results, improving the earlier packings were presented 2000 in Mátraháza, Hungary at the Nonlinear Optimization Winter School, and appeared in [7, 25]. Further results are published by Péter Gábor Szabó in [22, 23], and the Japanese versions are covered together with other similar problems in [34]. The paper [19] by Mihály Csaba Markót discusses a verified algorithm for the same problem class. In [26] he has first solved the packing of 28 circles case. The comment [26] corrects an earlier paper on circles packing. The improved packing for 47 circles is demonstrated on Figure 1. In [35] Péter Gábor Szabó discussed optimal substructures in optimal and approximate circle packings.

Mihály Csaba Markót wrote his PhD dissertation based on results in part on the field of interval optimization algorithm development, and in part on verifying the solutions of circle packing in the unit square problem instances of $n = 28, 29$ and 30 [25, 27]. The proven optimal solution for the $n = 28$ case is illustrated on Figure 2.

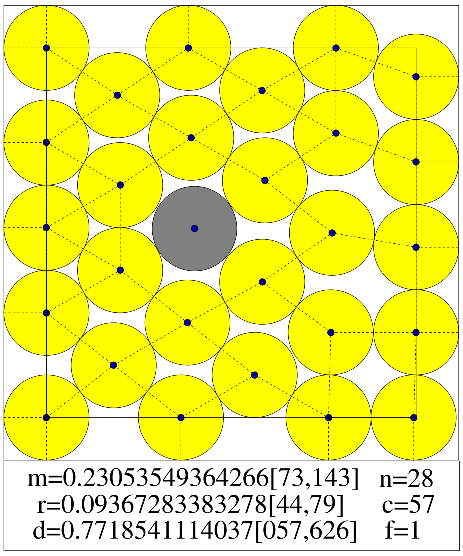

Figure 2. The verified optimal solution for the 28 circles packing problem. The notation is the same as on Figure 1.

III. OTHER OPTIMIZATION RELATED RESEARCH

János Balogh studied stochastic global optimization techniques applied successfully to a chemical phase stability problem in [2, 5]. The same author has published a paper on optimization on Stiefel manifolds with coauthors [1]. Boglárka Tóth and Tamás Vinkó tested a sophisticated interval arithmetic based algorithm on standard test problems. The results are published in [38]. András Erik Csallner investigated the consequences of Lipschitz continuity for the stopping rules in [1]. He have also investigated the method variant with the boxing algorithm with German coauthors [10].

The KÉSZ Ltd. asked us to contribute to its production control system with a cutting algorithm that improves the cutting of iron rods used in buildings. The research ended in the suggestion of a compound algorithm containing packing heuristics and enumeration procedures [14, 16] allowing a few percentages of savings in a very high yearly raw material budget.

IV. RESEARCH PROJECTS

Here we summarize the research and development projects of the last years won. In all but one cases Tibor Csendes was the (Hungarian) project head. For each project the source, identification number, time period, project title, and number of participating researchers are given.

OMFB Hungarian-German Bilateral, D-7/97, 1998-2000, "Interval arithmetic based global optimization", 4 persons.

FKFP 0449-99, 1999-2000, "New multisection techniques in global optimization", 2 persons.

OTKA F 025743, 1998-2001, "Optimization techniques", 2 persons (head of the project: András Erik Csallner).

OTKA T 032118, 2000-2003, "Efficient interval algorithms in nonlinear optimization", 1 person.

MTA Hungarian-Bulgarian collaboration, BGA-20, 2000-2004, "Development of New Efficient Optimization Methods for Chemical Engineering Simulation and Design Applications", 4 persons.

OTKA T 034350, 2001-2004, "Developments of global optimization procedures", 8 persons.

OMFB Hungarian-German Bilateral, D-30/00, 2001-2003, "Global optimization procedures", 7 persons.

MÖB-DAAD Hungarian-German, 11/01, "Application of global optimization methods in approximation problems", 7 persons.

KÉSZ Ltd. Szeged, "A cutting problem in metal industry", 2 persons.

OMFB Hungarian-Spanish Bilateral, SP-25/01, 2001-2003, "Reliable methods of global optimization and their parallel implementation", 11 persons.

OMAA Austrian-Hungarian Education and Research Cooperation Fund, 56ÖU11, Development of Global Optimization Methods and Their Application for Specific hard Problems, 4 persons.

V. CONFERENCES ORGANIZED

The authors organized three conferences in the interval 2000 – 2003. The first two are meant as the possibly first conference for many PhD students of computer science, it is the Conference for PhD Students in Computer Science (CS^2) held in Szeged, Hungary, 20-23 July, 2000 and 2-6 July, 2002. There were around 70 accepted talks from many countries, and around 80 participants. The next such conference will be held in Szeged again (1-4 July, 2004), the organization has already be began. The Acta Cybernetica, and the Periodica Polytechnica devoted special issues to the fully refereed papers emerging from the talks. The related information together with the lists of accepted papers and coordinates of the participants can be found at

`http://www.inf.u-szeged.hu/~cscs`

In the case of the third conference, Tibor Csendes was a member of the Organizing Committee of the EURO XVII conference, the main meeting of the European researchers in operations research. It was organized in Budapest, Hungary, July 16-19, 2000. The details can be found at

`http://www.sztaki.hu/conferences/euro17/`

Tibor Csendes edited special issues of the Central European Journal of Operations Research devoted to the papers emerging from the talks of the Hungarian Operations Research Conferences held in Veszprém, 1999 [24] with Tamás Rapcsák, and that held in Debrecen, 2001 [20] with Sándor Komlósi, and György Meszéna.

VI. CONNECTIONS OF RESEARCH TO EDUCATION

The above research relates mostly to the Global Optimization, Nonlinear Optimization and to the Interval Mathematics special courses and is also connected to the Global Optimization and Reliable Numerics PhD courses. The successful teaching methodology of interval mathematics is discussed in [11]. The open problems of some particular fields can serve as research subjects for student research competition (TDK) or for PhD students.

ACKNOWLEDGEMENTS

The mentioned research tasks were supported by the following grants: MTA Bulgarian-Hungarian Bilateral project BGA-20/2000, OMFB German-Hungarian Collaboration Grants D-30/00, OMFB Spanish-Hungarian Collaboration Grant SP-25/01, MÖB-DAAD collaboration Grant 11/01, OTKA T 016413, T 017241, F 025743,T 032118, and T 034350; FKFP 0449/99.

REFERENCES

[1] Balogh, J., T. Csendes, and T. Rapcsák: Some global optimization problems on Stiefel manifolds. Accepted for publication in J. Global Optimization

[2] Balogh J., T. Csendes, and R.P. Stateva: A fázis-stabilitás elemzése egy új célfüggvény és egy globális optimalizálási módszer fölhasználásával (In Hungarian), Magyar Kémiai Folyóirat 107(2001) 82-89

[3] Balogh, J., T. Csendes, and R.P. Stateva: Application of a stochastic method to the solution of the phase stability problem: cubic equations of state, Fluid Phase Equilibria 212(2003) 257-267.

[4] Casado, L.G., I. García, and T. Csendes: A new multisection technique in interval methods for global optimization. Computing 65(2000) 263-269

[5] Casado, L.G., I. García, and T. Csendes: A heuristic rejection criterion in interval global optimization. BIT 41(2001), 683-692

[6] Casado, L.G., I. García, T. Csendes, and V.G. Ruíz: Heuristic Rejection in Interval Global Optimization. JOTA 118(2003), 27-43

[7] Casado, L.G., I. García, P.G. Szabó, and T. Csendes: Equal circles packing in a square II. — new results for up to 100 circles using the TAMSASS-PECS algorithm. In: F. Giannessi, P. Pardalos, and T. Rapcsák (eds.): Optimization Theory: Recent Developments from Mátraháza, Kluwer, Dordrecht, 207-224, 2001

[8] Csallner A.E.: Lipschitz Continuity and the Termination of Interval Methods for Global Optimization. Computers and Mathematics with Application 42(2001) 1035-1042

[9] Csallner, A.E., T. Csendes, and M.CS. Markót: Multisection in Interval Methods for Global Optimization I. Theoretical Results, J. Global Optimization 16(2000) 371-392

[10] Csallner, A.E., Klatte, R., Ratz, and D., Wiethoff, A., Interval Methods for Global Optimization Using the Boxing Method, Proceedings of SCAN2000/Interval2000 Joint Conference, 205-214, 2001

[11] Csallner A.E., Mágoriné Huhn Á., and Balogh J., On the Teaching of Interval Computations, Proceedings of the 5th International Conference on Applied Informatics, 31-40, 2002

[12] Csallner, A.E., T. Csendes, and A.B. Kocsis: Reliable numerical computation in civil engineering. Accepted for publication in Numerical Algorithms.

[13] Csendes, T.: New subinterval selection criteria for interval global optimization. J. Global Optimization 19(2001) 307-327

[14] Csendes, T.: Interval Analysis: Subdivision Directions in Interval B&B Methods. In: Encyclopedia of Optimization, Kluwer, Dordrecht, 2001

[15] Csendes T.: Számítás garantált pontossággal I. (Computation with garantied reliability I, In Hungarian), Természet Világa, 132-134 (2001)

[16] Csendes T.: Számítás garantált pontossággal II., (Computation with garantied reliability II, In Hungarian), Természet Világa, 180-183 (2001)

[17] Csendes, T.: Numerical experiences with a new generalized subinterval selection criterion for interval global optimization. Reliable Computing, 9(2003) 109-125

[18] Csendes, T.: Generalized subinterval selection criteria for interval global optimization. Accepted for publication in Numerical Algorithms

[19] Csendes, T., R. Klatte, and D. Ratz: A Posteriori Direction Selection Rules for Interval Optimization Methods, CEJOR 8(2000) 225-236

[20] Csendes, T., S. Komlósi, and G. Meszéna (eds.): Special Issue: Proceedings of the XXV. Hungarian Operations Research Conference. CEJOR 11(2003) 1-101

[21] Csendes, T. and M. Markót: On a cutting stock problem of the metal industry (in Hungarian), KÉSZ Ltd., Szeged, 2000

[22] Csendes T. and M. Markót: On a cutting stock problem of the metal industry II (in Hungarian), KÉSZ Ltd., Szeged, 2000

[23] Csendes, T. and T. Rapcsák (eds.): Special Issue: Proceedings of the XXIV. Hungarian Operations Research Conference, CEJOR 8(2000) 1-92

[24] Markót, M.Cs.: An interval method to validate optimal solutions of the "packing circles in a unit square" problems. CEJOR 8(2000) 63-77

[25] Markót, M.Cs.: Garantált megbízhatóságú globális optimalizálási módszerek továbbfejlesztése korlátozásos feladatokra és alkalmazásuk körpakolási feladatok megoldása esetén (In Hungarian. English title: Reliable Global Optimization Methods for Constrained Problems and Their Application for Solving Circle Packing Problems). PhD dissertation. Szeged, 2003

[26] Markót, M.Cs.: Optimal Packing of 28 Equal Circles in a Unit Square - the First Reliable Solution. Accepted for publication in Numerical Algorithms

[27] M.C. Markót and T. Csendes: A new verified optimization technique for the "packing circles in a unit square" problems. Accepted for publication in SIAM J. Optimization.

[28] Markót, M.Cs., T. Csendes, and A.E. Csallner: Multisection in Interval Methods for Global Optimization II. Numerical Results, J. Global Optimization 16(2000) 219-228

[29] Markót, M.Cs., J. Fernandez, L.G. Casado, and T. Csendes: New interval methods for constrained global optimization. Accepted for publication in Mathematical Programming

[30] Martínez, J.A., L.G. Casado, I. García, Ya.D. Sergeyev, and B. Tóth, On an efficient use of gradient information for accelerating interval global optimization algorithms, accepted to Numerical Algorithms

[31] J.A. Martínez, L.G. Casado, I. García, and B. Tóth. AMIGO: Advanced Multidimensional Interval analysis Global Optimization algorithm. To appear in book series of Nonconvex Optimization and Its Applications, Kluwer Academic Publishers. Devoted to the 4th International Conference on Frontiers in Global Optimization.

[32] Szabó, P.G.: Some new structures for the "equal circles packing in a square" problem. CEJOR 8(2000) 79-91

[33] Szabó P.G.: Optimális körelhelyezések a négyzetben (In Hungarian), Polygon 10(2000) 48-64

[34] Szabó P.G.: Sangaku, matematikai fatáblák japán templomokban, Középiskolai Matematikai és Fizikai Lapok (In Hungarian), 7:386-388, 2001

[35] Szabó, P.G., Optimal substructures in optimal and approximate circle packings, Beiträge zur Algebra und Geometrie, accepted for publication

[36] Szabó, P.G. and T. Csendes: Kiegészítés Filep László tanulmányához. (In Hungarian). Magyar Tudomány, 8(2001) 984-985

[37] Szabó, P.G., T. Csendes, L.G. Casado, and I. García: Equal circles packing in a square I. — problem setting and bounds for optimal solutions. In: F. Giannessi, P. Pardalos, and T. Rapcsák (eds.): Optimization Theory: Recent Developments from Mátraháza, Kluwer, Dordrecht, 191-206, 2001

[38] Tóth, B. and T. Vinkó: Egy hatékony számítógépes eszköz matematikai problémák megoldására (In Hungarian), Polygon, (11)2002 19-42

[39] Vinkó, T., J.-L. Lagouanelle, and T. Csendes: A New Inclusion Function for Optimization: Kite — The One Dimensional Case. J. Global Optimization, accepted for publication

[40] Tamás Vinkó, Dietmar Ratz: A Multidimensional Branch-and-Prune Method for Interval Global Optimization. Accepted for publication in Numerical Algorithms

[41] Vladik Kreinovich and Tibor Csendes: Theoretical Justification of a Heuristic Subbox Selection Criterion for Interval Global Optimization. CEJOR 9(2001) 255-265

ANALYSIS OF ALGORITHMS

János Csirik and Csanád Imreh

I. THE CLASSICAL BIN PACKING PROBLEM

In the classical bin packing, we are given a list L of items $(a_1, a_2, ..., a_n)$ each item $a_i \in (0,1]$ and the goal is to find a packing of these items into a minimum number of unit-capacity bins.

In [4] we present a sequence of new linear-time, bounded-space, on-line bin packing algorithms, the K -Bounded Best Fit algorithms (BBF_K). They are based on the $\Theta(nlogn)$ Best Fit algorithm in much the same way as the Next-K Fit algorithms are based on the $\Theta(nlogn)$ First Fit algorithm. Unlike the Next-K Fit algorithms, whose asymptotic worst-case ratios approach the limiting value of $\frac{17}{10}$ from above as $K \to \infty$ but never reach it, these new algorithms have worst-case ratio $\frac{17}{10}$ for all $K \geq 2$. They also have substantially better average performance than their bounded-space competition, as we have determined based on extensive experimental results for instances with item sizes drawn independently and uniformly from intervals of the form $(0,u], 0 < u \leq 1$. Indeed, for each $u < 1$, it appears that there exists a fixed memory bound $K(u)$ such that $BBF_{K(u)}$ obtains significantly better packings on average than does the First Fit algorithm, even though the latter requires unbounded storage and has a significantly greater running time. For $u = 1$, BBF_K can still outperform First Fit (and essentially equal Best Fit) if K is allowed to grow slowly. We provide both theoretical and experimental results concerning the growth rates required.

In [8] we study online bounded space bin packing in the resource augmentation model of competitive analysis. In this model, the online bounded space packing algorithm has to pack a list L of items in (0,1] into a small number of bins of size $b \geq 1$. Its performance is measured by comparing the produced packing against the optimal offline packing of the list L into bins of size 1. We present a complete solution to this problem: For every bin size $b \geq 1$, we design online bounded space bin packing algorithms whose worst case ratio in this model comes arbitrarily close to a certain bound $\rho(b)$. Moreover, we prove that no online bounded space algorithm can perform better than $\rho(b)$ in the worst case.

In [7] and [6] we present experimental and theoretical results of the deterministic on-line Sum-of-Squares algorithm. SS is applicable to any instance of bin packing in which the bin capacity B and the item sizes $s(a)$ are integer (or can be scaled to be so). It performs remarkably well from an average case point of view: for any discrete distribution in which the optimal expected waste is sublinear, SS also has sublinear expected waste. For any discrete distribution where the optimal expected waste is bounded, SS has expected waste at most $O(logn)$. In addition, we present a randomized $O(nBlogB)$ -time on-line algorithm SS^* based on SS, whose expected behavior is essentially optimal for all discrete distributions. Algorithm SS^* also depends on a new linear-programming based pseudopolynomial-time algorithm for solving the NP-hard problem of determining, given a discrete distribution F, just what is the growth rate for the optimal expected waste. An off-line randomized variant SS^{**} performs well in a worst-case sense: For any list L of integer-sized items to be packed into bins of a fixed size B, the expected number of bins used by SS^{**} is at most $OPT(L) + \sqrt{OPT(L)}$.

In [10] a modified version of the online strip packing problem is presented. In the strip packing problem the goal is to pack a set of rectangles into a vertical strip so as to minimize the total height of the strip needed. In the modified version it is allowed to change the form of the rectangles by lengthening them, keeping the area fixed. For the solution of the problem a 4-competitive algorithm is given and we prove that there is no algorithm which has a smaller competitive ratio than 1.73.

In [2] an overview of the approximation algorithms for the bin packing problem is given.

II. BIN COVERING

Bin covering takes as input a list of item sizes and places them into bins of unit demand so as to maximize the number of bins whose demand is satisfied. This is in a sense a dual problem to the classical one-dimensional bin packing problem, but has for many years lagged behind the latter as far as the quality of the best approximation algorithms. In [5] we design algorithms to close the gap, both in terms of worst-case and average-case results. We present (1) the first approximation scheme for the offline version, (2) algorithms with bounded worst-case behavior whose expected behavior is asymptotically optimal for all discrete "perfect-packing distributions" (ones for which optimal packing has sublinear expected waste), and (3) a learning algorithm that has asymptotically optimal expected behavior for all discrete distributions. The algorithms of (2) and (3) are based on the recently-developed Sum-of-Squares algorithm [6] for bin packing, whose adaptation to the case of bin covering requires new ideas and analytical techniques.

III. SCHEDULING

In [1] we consider the problem of scheduling a sequence of tasks in a multi-processor system with conflicts. Conflicting processors cannot process tasks at the same time. At certain times new tasks arrive in the system, where each task specifies the amount of work (processing time) added to each processor's workload. Each processor stores this workload in its *input buffer*. Our objective is to schedule task execution, obeying the conflict constraints, and minimizing the maximum buffer size of all processors. In the off-line case, we prove that, unless $P = NP$, the problem does not have a polynomial-time algorithm with a polynomial approximation ratio. In the on-line case, we provide the following results: (i) a competitive algorithm for general graphs, (ii) tight bounds on the competitive ratios for cliques and complete k-partite graphs, and (iii) a $(\Delta/2 + 1)$-competitive algorithm for trees, where Δ is the diameter. We also provide some results for small graphs with up to 4 vertices.

In [11] and [12] the following scheduling problem is investigated. There are two sets of identical machines, the jobs have two processing times one for each set of machines. We consider two different objective functions, in the first model the goal is to minimize the maximum of the makespans on the sets, in the second model we minimize the sum of the makespans. In [11] we investigate a semi online version of the first model. In this version the decision maker has to decide in online fashion where to schedule the jobs, but it can produce an offline preemptive schedule at the end. In the paper a $1 + \sqrt{2}$-competitive algorithm is presented for arbitrary k and m, furthermore a general lower bound of 1.781 is proven for the possible competitive ratios. In [12] we investigate the online and the offline versions. In the offline case an FPTAS is given for both objective functions. In the online case we characterize the competitive ratio of a load greedy algorithm which assigns every job to the set, where the job has smaller load. The competitive ratio of this algorithm tends to ∞ for both problems as the ratio m/k grows. We show that the post greedy algorithm has also unbounded competitive ratio as m/k grows. Finally we analyze a modified greedy algorithm and we show that it is constant competitive for arbitrary number of machines. Lower bound for the possible competitive ratios are also presented.

IV. PAGING

In [3] we consider a variant of the online paging problem where the online algorithm may buy additional cache slots at a certain cost. The overall cost incurred equals the total cost for the cache plus the number of page faults. This problem and our results are a generalization of both, the classical paging problem and the ski rental problem. We derive the following three tight results: (1) For the case where the cache cost depends linearly on the cache size, we give a λ-competitive online algorithm where $\lambda \approx 3.14619$ is a solution of $\lambda = 2 + ln\lambda$. This competitive ratio λ is best possible. (2) For the case where the cache cost grows like a polynomial of degree d in the cache size, we give an online algorithm whose competitive ratio behaves like $d/lnd + o(d/lnd)$. No online algorithm can reach a competitive ratio better than d/lnd. (3) We exactly characterize the class of cache cost functions for which there exist online algorithms with finite competitive ratios.

V. SERVER PROBLEMS

In [9] we consider a generalized 2-server problem on the uniform space in which servers have different costs. Previous work focused on the case where the ratio between these costs was very large. We give results for varying ratios. For ratios below 2.2, we present a best possible algorithm which is trackless. We present a general lower bound for trackless algorithms depending on the cost ratio, proving that our algorithm is the best possible trackless algorithm up to a constant factor for any cost ratio. The results are extended for the case where we have two sets of servers with different costs.

REFERENCES

[1] M. Chrobak, J. Csirik, Cs. Imreh, J. Noga, J. Sgall, and G. J. Woeginger. The buffer minimization problem for multiprocessor scheduling with conflicts. In *Automata, Languages and Programming, Lecture Notes in Computer Science, Springer, Vol. 2076*, pages 862–874, 2001.

[2] E. G. Coffman, J. Csirik, G. Woeginger Approximate solutions to bin packing problems. In *Handbook of Applied Optimization* (Eds. P. M. Pardalos, M. C. G. Resende), Cambridge University Press, pages 607-615, 2002.

[3] J. Csirik, Cs. Imreh, J. Noga, S. S. Seiden, and G. J. Woeginger. Buying a constant competitive ratio for paging. In *9th Annual European Symposium, Lecture Notes in Computer Science, Springer, Vol. 2161*, pages 98–108, 2001.

[4] J. Csirik and D. S. Johnson. Bounded space on-line bin packing: Best is better than first. *Algorithmica*, 31:115–138, 2001.

[5] J. Csirik, D. S. Johnson, and C. Kenyon. Better approximation algorithms for bin covering. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 557–566. ACM Press, 2001.

[6] J. Csirik, D. S. Johnson, C. Kenyon, J. B. Orlin, P. W. Shor, and R. R. Weber. On the sum-of-squares algorithm for bin packing. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 208–217. ACM Press, 2000.

[7] J. Csirik, D. S. Johnson, C. Kenyon, P. W. Shor, and R. R. Weber. A self organizing bin packing heuristic. In *Algorithm Engineering and Experimentation, ALENEX '99, Lecture Notes in Computer Science, Vol. 1619, Springer,* , pages 246–265, 1999.

[8] J. Csirik and G. Woeginger. Resource agmentation for online bounded space bin packing. In *Automata, Languages and Programming, Lecture Notes in Computer Science, Vol. 1853, Springer*, pages 296–304, 2000.

[9] L. Epstein, Cs. Imreh, R. van Stee. More on Weighted Servers or FIFO is better than LRU. *Teorethical Computer Science*, 306:305–317, 2003.

[10] Cs. Imreh. Online strip packing with modifiable boxes. *Operations Research Letters*, 29:79–86, 2001.

[11] Cs. Imreh. An online scheduling algorithm for a two-layer multiprocessor architecture. *Acta Cybernetica* 15:163–172, 2001.

[12] Cs. Imreh. Scheduling problems on two sets of identical machines. *Computing*, 70:277–294, 2003.

RESEARCH IN THEORETICAL COMPUTER SCIENCE

Zoltán Ésik

I. ITERATION THEORIES AND THE AXIOMATIZATION OF THE FIXED POINT OPERATION IN COMPUTER SCIENCE

The purpose of the work carried out from 2000 to 2004 was to obtain complete, but simple, descriptions of the properties of the fixed point or iteration operation in computation. Previous work by Bloom, Ésik and others has resulted in a complete description of the equational laws satisfied by the iteration operation. This description takes the form of the axioms for *Iteration Theories* [10], certain (enriched) Lawvere algebraic theories, which capture important features of many classes of structures of interest in the theory of computation involving ordered and metric structures, functors and 2-categories, trees and synchronization trees, and others. In [12], we have introduced iteration 2-theories that are a 2-categorical generalization of iteration theories. We have described the structure of the free iteration 2-theories.

One of the axiom systems of iteration theories involves an identity associated with each finite group. In [26], we studied the independence of the groups identities and described the structure of the free theories satsifying the Conway equations and any subcollection of the group identities. In [11], we have shown that iteration theories cannot be axiomatized by a finite number of equation schemes. In [27], we described all iteration theories of boolean functions.

II. APPLICATIONS OF THE EQUATIONAL LOGIC OF FIXED POINTS

As major beneficial corollaries of the general theory of fixed points, we have obtained relatively simple sets of equational and implicational axioms for the binary supremum operation on continuous functions [28] and the concurrent behavior of processes with respect to several behavioral equivalences [28, 32, 7]. For a survey, see [31].

Using only equations involving fixed points, we established several Kleene-type theorems for power series, tree series, semi-additive categories [39, 15, 13]. For a survey on tree series, see [45].

We gave pure equational proofs of some fundemantal results of language theory including Parikh's theorem and the Greibach normal form theorem for context-free languages, cf. [6, 47, 48].

III. SEMIRINGS

In [41], we have defined inductive *-semirings that are partially ordered semirings with a star operaion satifying Park's induction axiom. We have established several basic properties of inductive *-semirings and related them to the Kleene algebras of Kozen. In [42], we studied a subclass of inductive *-semirings in which certain infinite sums exists. We have completely described the structure of free rationally additive semirings. Several basic properties of locally closed semirings have been studied in [43].

IV. TROPICAL SEMIRINGS

In a series of papers, we studied the equational theory of the tropical semirings. We established several axiomatizability and decidability results and described the structure of the free algebras in the corresponding varieties. See [4, 1, 2, 3, 5].

V. AXIOMATIZING REGULAR LANGUAGES AND RATIONAL POWER SERIES

In [22], we proved that the equational theory of regular languages with concatenation and Kleene star is nonfinitely based. This solves a problem of Bredikhin. In [40], we extended Kozen's axiomatization of the equational theory of regular languages to rational power series with coefficients in any finite commutative ordered semiring. In [21], we proved that the equational theory of Kleene algebras of binary relations with relational inverse is not finitely based.

VI. THE SHUFFLE OPERATION AND POMSET MODELS OF CONCURRENCY

In earlier work, we extended the study of the equational theory of the regular operations on languages to include the shuffle operation which, in the interleaving model of concurrency, is used to model parallel execution of processes. We showed that the equational theory of the shuffle operation is intimately related to the theory of pomsets equipped with variants of the series and parallel composition operations. In [33], we studied equational theory of the true concurrency model together with some behavioral preorders.

VII. STRUCTURE THEORY OF AUTOMATA

In [25], we gave a simple proof of the Krohn-Rhodes decomposition theorem for finite automata. In [24], we have shown that there is a Letichevsky automaton not complete with respect to homomorphic representation by the ν_2-product. (On the other hand, it is known that every Letichevsky automaton is complete with respect to homomorphic representation by the ν_3-product.) We have obtained a similar result for the homomorphic simulation, cf. [23].

VIII. HIGHER DIMENSIONAL AUTOMATA

In [49, 50], continuing the work started in [29], we defined automata on higher dimensional words and extended several basic results of the theory of automata to this setting including the equivalence to algebraic recognizability and monadic second-order definability. The results were obtained in collaboration with Z. L. Németh, Department of Foundations of Computer Science.

IX. ALGEBRAS FOR ASYNCHRONOUS CIRCUITS

We have proposed a family of algebras to study hazards in asynchronous circuits, cf. [18, 19, 20]. These algebras are commutative De Morgan bisemigroups and are a generalization ternary algebras. In [30], we gave a geometric description of the free (commutative) De Morgan bisemigroups and De Morgan bisemilattices.

X. REGULAR WORDS

Regular words are those words isomorphic to the frontier of regular trees. We gave a polynomial time algorithm to decide whether a regular word is scattered, cf. [14]. Regular words are natuarally equipped with the "scattered operations" of concatenation, omega power and its opposite; and the "shuffle operations". In [16], we solved a longstanding open problem of B. Courcelle by providing a set of simple equational axioms for the scattered operatuions. We also showed the nonexistence of a finite axiomatization. In [17], we gave equational axioms for both the scattered and the shuffle operations. Moreover, we proved that the equational theory is decidable in polynomial time.

XI. LOGIC AND AUTOMATA

In [38], we have extended first-order logic and linear temporal logic on finite words with a sort of modular counting. We have charaterized the expressive power of these logics by an extension of the notion of aperiodicity. In [46], we have associated a generalized quantifier with any regular word language and used the double semidirect product of finite generated monoids to describe the expressive power of the resulting logics. In [34], we have studied a related extension of linear temporal logic using semidirect prodicts of finite generated monoids and obtained several algebraic characterizations. Some of the results of [46] have been extended in [51] to finite trees using the newly introduced notion of double semidirect products of preclones.

XII. CONFERENCES AND EDITORIAL WORK

In addition to a large number of other conference presentations, Zoltán Ésik was invited speaker at the following workshops and conferences: Words, Languages and Combinatorics (Kyoto, 2000), Category Theory (Como, 2000), Developments in Language Theory (Vienna, 2001), Fixed Points in Computer Science (Florence, 2001), Workshop on Max-Plus Algebras (Prague, 2001), Expressiveness in Concurrency (Aalborg, 2001), 6th Int. Conf. Discrete Mathematics and Applications (Banska, 2001), Developments in Language Theory (Kyoto, 2002), Workshop on Weighted Automata and Applications (Dresden, 2002), Workshop on Automata, Concurrency, and Logic (Madras, 2003), Coalgebraic Methods in Computer Science (Warsaw, 2003). He served on the steering committee of the Fundamentals of Computer Science conferences and on the program committee of the workshops and conferences Fixed Points in Computer Science (Paris, 2000), Fixed Points in Computer Science (Florence, 2001), Fixed Points in Computer Science (Copenhagen, 2002, co-chairman), Automata and Formal Languages (Debrecen, 2002), Developments in Language Theory (Szeged, 2003, co-chairman), Fundamentals of Computation Theory (Malmö, 2003). Fixed Points in Computer Science (Warsaw, 2003, co-chairman), Expressiveness in Concurrency (Marseille, 2003), Process Algebra: Open Problems and Future Directions, (Bertinoro, 2003). Zoltán Ésik co-edited the proceedings of FICS 02, FICS 03, DLT 03, Process Algebra: Open Problems and Future Directions 2003, and served as an editor of the journals Theoretical Computer Science, Theoretical Informatics and Applications, Discrete Mathematics and Theoretical Computer Science, Acta Cybernetica. He was a member of TC1 of International Federation for Information Processing. In 2003, he was elected to the Council of the European Association for Theoretical Computer Science. He held visiting positions in Aizu, Bordeaux, Paris, Kyoto, Waterloo, Aalborg, Dresden.

ACKNOWLEDGMENTS

The above research has been supported by various grants from the National Foundation of Hungary for Scientific Research, the Ministry of Education of Hungary, the National Science Foundation, the Japan Society for the Promotion of Science, the Humboldt Foundation, the Austrian-Hungarian Action Foundation, the French-Hungarian Bilateral Fund BALATON, and the Austrian-Hungarian Research and Development Program.

REFERENCES

[1] L. Aceto, Z. Ésik and A. Ingólfsdóttir, On the two-variable segment of the equational theory

of the max-sum algebra of natural numbers, in: proc. *STACS 2000, Lille*, LNCS 1770, Springer-Verlag, 2000, 267–278

[2] L. Aceto, Z. Ésik and A. Ingólfsdóttir, Axiomatizing tropical semirings, in: *Foundations of Software Science and Computation Structures, Genova, 2001*, F. Honsell and M. Miculan editors), Lecture Notes in Computer Science 2030, Springer-Verlag, April 2001, 42–56.

[3] L. Aceto, Z. Ésik and A. Ingólfsdóttir, Non-finitely Based Tropical Semirings, in: Max-plus Algebras and Their Applications to Discrete-event Systems, Theoretical Computer Science, and Optimization, 27-29 August, 2001, Prague, (S. Gaubert and J. J.Loiseau editors), IFAC (International Federation of Automatic Control) Publications, Elsevier Science, 2001, pp. 29–34.

[4] L. Aceto, Z. Ésik and A. Ingólfsdóttir, The max-sum algebra of natural numbers has no finite equational basis, *Theoretical Computer Science*, 293(2003), 169–188.

[5] L. Aceto, Z. Ésik and A. Ingólfsdóttir, Equational theories of tropical semirings, *Theoretical Computer Science*, 298(2003), 417–469.

[6] L. Aceto, Z. Ésik and A. Ingólfsdóttir, A fully equational proof of Parikh's theorem, *Theoretical Informatics and Applications*, 36(2002), 129–154.

[7] L. Aceto, Z. Ésik and A. Ingólfsdóttir, Equational axioms for probabilistic bisimilarity, in: Proceedings of *Algebraic Methodology and Software Technology, AMAST 2002*, Reunion, LNCS 2422, Springer, 2002, 239–253.

[8] L. Aceto, Z. Ésik, W. J. Fokkink, A. Ingólfsdóttir, Eds., Process Algebra: Open Problems and Future Directions, PA '03, Bologna, Italy, 21–25 July 2003, BRICS Notes Series, NS 03-3, 2003 .

[9] S.I. Aleshnikov, Ju. F. Boltnev, Z. Ésik, S.A. Ishanov and W. Kuich, Formal languages and automata I: Conway semirings and finite automata (in Russian), Vectnik, Kaliningradckogo gocudarctvennogo universiteta, Seria Informatika and telekommunikacia, 3(2003), 7–38.

[10] S.L. Bloom and Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science, Springer–Verlag, 1993.

[11] S.L. Bloom and Z. Ésik, There is no finite axiomatization of iteration theories, in: proc. *LATIN 2000, Punta del Este, Uruguay*, LNCS 1776, Springer-Verlag, 2000, 367–376.

[12] S.L. Bloom, Z. Ésik, A. Labella and E.G. Manes, Iteration 2-theories, *Applied Categorical Structures*, 9(2001), 173–216.

[13] S.L. Bloom and Z. Ésik, Unique guarded fixed-points in an additive setting, in: Proceedings of *Category Theory and Computer Science*, CTCS 2002, Ottawa, ENTCS, vol. 69, Elsevier, 2002 16 pages (electonic).

[14] S.L. Bloom and Z. Ésik, Deciding whether the frontier of a regular tree is scattered, *Fundamenta Informaticae*, 55(2003), 1–21.

[15] S.L. Bloom and Z. Ésik, An extension theorem with an application to formal tree series, *J. of Automata, Languages and Combinatorics,* 8(2003), 145–185.

[16] S.L. Bloom and Z. Ésik, Axiomatizing omega and omega-op power on words, *Theoretical Informatics and Applications*, to appear.

[17] S.L. Bloom and Z. Ésik, Axioms for regular words, in: *Proc. FST&TCS 03, Mumbai*, LNCS 2914, Springer, 2003, 50–61.

[18] J. Brzozowski and Z. Ésik, Hazard algebras, in: *Half Century of Automata Theory*, London, ON, 2000, World Scientific, Singapore, 2001, 1–19.

[19] J. Brzozowski and Z. Ésik, Hazard algebras, *Formal Methods in System Design*, 23(2003), 223–256.

[20] J. Brzozowski, Z. Ésik and Y. Iland, Algebras for hazard detection, in: *31st IEEE International Symposium on Multiple-Valued Logic*, 22-24 May 2001, Warsaw, Poland, IEEE Computer Society, Los Alamitos, California, 2001, 3–12.

[21] S. Crvenković, I. Dolinka and Z. Ésik, The variety of Kleene algebras with conversion is not finitely based, *Theoretical Computer Science*, 230(2000), 235–245.

[22] S. Crvenković, I. Dolinka and Z. Ésik, On equations for union-free regular languages, *Information and Computation*, 164(2001), 152–172.

[23] P. Dömösi and Z. Ésik, Homomorphic Simulation and Letichevsky's Criterion, *J. of Automata, Languages and Combinatorics,* 6(2001), 427–436.

[24] P. Dömösi and Z. Ésik, A note on completeness of the ν_3-product, *Publ. Math.*, 60(2002), 539–550.

[25] Z. Ésik, A new proof of the Krohn–Rhodes decomposition theorem, *Theoretical Computer Science*, 234(2000), 287–300.

[26] Z. Ésik, The power of the group identities for iteration, *Int. J. Algebra and Computation*, 10(2000), 349–373.

[27] Z. Ésik, Iteration theories of boolean functions, in: proc. *Math. Found. of Computer Science, Bratislava, 2000*, LNCS 1893, Springer-Verlag, 2000, 343–352.

[28] Z. Ésik, Axiomatizing the least fixed point operation and binary supremum, in: proc. *Computer Science Logic, Munich, 2000*, LNCS 1862, Springer-Verlag, 302–316.

[29] Z. Ésik, Free algebras for generalized automata and language theory, in: proc. *Algebraic Systems, Formal Languages and Computation, RIMS Kokyuroku 1166*, Kyoto University, 2000, 52–58.

[30] Z. Ésik, Free De Morgan bisemigroups and bisemilattices. *Algebra Collquium*, 10(2003), 23–32.

[31] Z. Ésik, The equational theory of fixed points with applications to generalized language theory, *Developments in Language Thoery*, Wien, July 2001, LNCS 2295, Springer-Verlag, 2002, p. 21–36.

[32] Z. Ésik, Continuous additive algebras and injective simulations of synchronization trees, *J. Logic. Comput.*, 12(2002), 271–300.

[33] Z. Ésik, Axiomatizing the subsumption and subword preorders on finite and infinite partial words, *Theoretical Computer Science*, 273(2002), 225–248.

[34] Z. Ésik, Extended temporal logic on finite words and wreath product of monoids with distinguished generators, in: *Proc. DLT 02, Kyoto*, LNCS 2450, 43–58, Springer, 2003.

[35] Z. Ésik and Z. Fülöp, Eds., Developments in Language Theory, 7th International Conference, DLT 2003, LNCS 2710, Springer, 2003.

[36] Z. Ésik and A. Ingólfsdóttir, Eds., Fixed Points in Computer Science 02, Copenhagen, Proceedings, BRICS Notes Series, NS-02-2, 2002.

[37] Z. Ésik and A. Ingólfsdóttir, Eds., Fixed Points in Computer Science 02, Copenhagen, Special issue of Theoretical Informatics and Applications, 37(2003), 271–391.

[38] Z. Ésik and M. Ito, Temporal Logic with Cyclic Counting and the Degree of Aperiodicity of Finite Automata, *Acta Cybernetica*, 16(2003), 1–28.

[39] Z. Ésik and W. Kuich, A Kleene theorem for Lindenmayerian algebraic power series, *J. of Automata, Languages and Combinatorics,* 5(2000), 109–122.

[40] Z. Ésik and W. Kuich, A generalization of Kozen's axiomatization of the equational theory of the regular sets, in: *Words, Semigroups, and Transductions*, World Scientific, 2001, 99–114.

[41] Z. Ésik and W. Kuich, Inductive *-semirings, *Theoret. Comput. Sci*, to appear.

[42] Z. Ésik and W. Kuich, Rationally additive semirings, J. Universal Computer Science, 2(8) 2002, 173–183.

[43] Z. Ésik and W. Kuich, Locally closed semirings, *Monatshefte Mathematik*, 137(2002), 21–29.

[44] Z. Ésik and W. Kuich, Conway-Halbringe als Grundlage für eine mathematische Automatentheorie. In: Doklady meschdunarodnogo matematitscheskogo seminara k 140-letiju so dnja roschdenija Davida Gilberta iz Kenigsberga i 25-letiju matematitscheskogo fakulteta (Vorträge des internationalen mathematischen Seminars zum 140. Geburtstag David Hilberts aus Königsberg und zum 25 - jährigen Jubiläum der mathematischen Fakultät), (S.I. Aleschnikov, S.Ju. Piljugin, Ju.I. Schevtschenko, Herausgeber), Universität in Königsberg, 2002, 240-246.

[45] Z. Ésik and W. Kuich, Formal tree series, *J. of Automata, Languages and Combinatorics,* 8(2003), 219–285.

[46] Z. Ésik and K.G. Larsen, Regular languages defined by Lindstöm quantifiers, *Theoretical Informatics and Applications*, 37(2003), 179–242.

[47] Z. Ésik and H. Leiss, Greibach normal form in algebraically complete semirings, in: Proceedings of *Computer Science Logic, CSL 2002*, Edingurgh, LNCS 2471, Springer, 2002, 135–150.

[48] Z. Ésik and H. Leiss, Algebraically complete semirings and Greibach normal form, *Annals of Pure and Applied Logic*, to appear.

[49] Z. Ésik and Z. Németh, Automata on series-parallel biposets, in: *Developments in Language Theory*, 5th Int. Conf., LNCS 2295, Springer, 2002, 217–227.

[50] Z. Ésik and Z. Németh, Higher dimensional automata, *J. of Automata, Languages and Combinatorics,* to appear.

[51] Z. Ésik and P. Weil, On logically defined recognizable tree languages, in: *Proc. FST&TCS 03, Mumbai*, LNCS 2914, Springer, 2003, 195–106.

REVERSE ENGINEERING

Rudolf Ferenc, Árpád Beszédes, László Vidács, and Tibor Gyimóthy

I. INTRODUCTION

Reverse engineering is "the process of analyzing a subject system to (a) identify the system's components and their interrelationships and (b) create representations of a system in another form at a higher level of abstraction". We refer to the extracted information as *facts* about the software system. The form of the extracted facts in terms of a set of entities with attributes and relationships is described by *schemas*. A *schema instance* is an incarnation of the schema which models a concrete software system.

We developed a process and a framework to facilitate fact extraction (see Section II.). The extracted facts conform to well-defined schemas to promote tool interoperability (see Section III.).

II. FACT EXTRACTION FROM C/C++ SOURCE CODE

We define *fact extraction* as a process which determines different steps that describe the way how facts about the source code can be obtained. These steps include acquiring project/configuration information, the analysis of the source files with analyzer tools, the creation of some kind of representations of the extracted facts, the merging of these representations and different processings made on this merged representation to enable the actual use of the collected information, like calculating metrics and mining design patterns (See Section IV.).

We developed a reverse engineering framework called *Columbus* [3] in cooperation with the Nokia Research Center in Helsinki. The main motivation behind developing the framework was to create a toolset which supports fact extraction and provides a common interface for other reverse engineering tasks as well.

The Columbus framework produces facts according to our schemas for the C/C++ language (see Section III.).

III. SCHEMAS

Schemas play a very important role in the process of fact extraction. They define the central repository of the whole process where the facts are stored. We designed two schemas that prescribe the form for storing the facts: the *Columbus Schema for C++ Preprocessing* (for preprocessing related facts) [6] and the *Columbus Schema for C++* (for the C++ language itself) [2, 3, 5].

To make the results of fact extraction widely usable, we further process the schema instances to take various new formats. These can be very simple transformations, like for instance XML and HTML, but the processing can be much more sophisticated, like calculating metrics and recognizing design patterns. We do also a so-called code audit processing for checking the source code against a set of coding rules.

IV. MINING DESIGN PATTERNS

Our algorithm discovers design pattern instances in the built-up schema instances. It introduces a new approach to the problem of pattern detection which includes the detection of operation call delegations, object creations and operation redefinitions. These elements identify pattern occurrences more precisely. The pattern descriptions are stored in an XML-based format designed by us, the Design Pattern Markup Language (DPML). This gives the freedom to modify the patterns, adapt them to specific needs, or create new pattern descriptions [1, 4].

CONFERENCE ORGANIZATION AND EDITORIAL WORK

The *European Conference on Software Maintenance and Reengineering* (CSMR2002) held in Budapest in 2002 was organized by people from this department. The most significant software maintenance conference, the *International Conference on Software Maintenance* (ICSM2005) will also be organized by this department in 2005.

Tibor Gyimóthy is a co-editor of the special issue of the *Journal of Software Maintenance and Evolution* for CSMR 2003.

REFERENCES

[1] Z. Balanyi and R. Ferenc. Mining Design Patterns from C++ Source Code. In *Proceedings of the 19th International Conference on Software Maintenance (ICSM 2003)*, pages 305–314. IEEE Computer Society, Sept. 2003.

[2] R. Ferenc and Á. Beszédes. Data Exchange with the Columbus Schema for C++. In *Proceedings of the 6th European Conference on Software Maintenance and Reengineering (CSMR 2002)*, pages 59–66. IEEE Computer Society, Mar. 2002.

[3] R. Ferenc, Á. Beszédes, M. Tarkiainen, and T. Gyimóthy. Columbus – Reverse Engineering Tool and Schema for C++. In *Proceedings of the 18th International Conference on*

Software Maintenance (ICSM 2002), pages 172–181. IEEE Computer Society, Oct. 2002.

[4] R. Ferenc, J. Gustafsson, L. Müller, and J. Paakki. Recognizing Design Patterns in C++ programs with the integration of Columbus and Maisa. *Acta Cybernetica*, 15:669–682, 2002.

[5] R. Ferenc, S. E. Sim, R. C. Holt, R. Koschke, and T. Gyimóthy. Towards a Standard Schema for C/C++. In *Proceedings of the 8th Working Conference on Reverse Engineering (WCRE 2001)*, pages 49–58. IEEE Computer Society, Oct. 2001.

[6] L. Vidács, Á. Beszédes, and F. Rudolf. Columbus Schema for C/C++ Preprocessing. In *Proceedings of the 8th European Conference on Software Maintenance and Reengineering (CSMR 2004)*, to appear. IEEE Computer Society, Mar. 2004.

TREE PROCESSING DEVICES

Zoltán Fülöp

I. TREE TRANSDUCERS

In [1] we prove that the domains of partial attributed tree transducers are the tree languages recognized by tree walking automata in universal acceptance mode. Moreover, we prove that the domains of partial attributed tree transducers having monadic output are the tree languages recognized by deterministic tree walking automata.

In [3] we consider attributed tree transducers of type simple multi-visit, multi-sweep, multi-alternating pass, and multi-pass following the definition of the same subclasses of attribute grammars. We characterize the tree transformation classes induced by the considered four subclasses of attributed tree transducers in terms of macro tree transducers. Namely, we define simple multi-visit, multi-sweep, multi-alternating pass and multi-pass macro tree transducers and prove formally that the attributed tree transducers and the macro tree transducers of the same type induce the same tree transformation class. Also we give an inclusion diagram of the tree transformation classes induced by the above and some further fundamental types of attributed and of macro tree transducers.

In [5] we consider the closure $UCI(Rel)$ of the class of relabelling tree transformations, under U=union, C=composition and I=iteration. We give a characterization of $UCI(Rel)$ in terms of a short expression built up from Rel with composition and iteration. We also give a characterization of $UCI(Rel)$ in terms of one-step rewrite relations of very simple term rewrite systems. We give a similar characterization of $UC(FRel_+)$ where $FRel_+$ is the class consisting of the transitive closures of all functional relabelling tree transformations. Finally we show that $UCI(Rel) = UCI(FRel)$.

In [7] we deal with shape-preserving top-down tree transducers. As top-down tree transducers generalize GSM's, shape preserving top-down tree transducers naturally generalize length preserving GSM's. For instance, top-down finite state relabelling tree transducers are shape preserving top-down tree transducers. We show that a top-down tree transducer is shape preserving if and only if it is equivalent to a top-down finite state relabelling tree transducer. We also prove that it is decidable if a top-down tree transducer is shape preserving.

II. TREE SERIES TRANSDUCERS

In [6] we generalize bottom-up tree transducers and top-down tree transducers to the concept of *bottom-up tree series transducer* and *top-down tree series transducer*, respectively, by allowing formal tree series as output rather than trees, where a formal tree series is a mapping from output trees to some semiring. We associate two semantics with a tree series transducer: a mapping which transforms trees into tree series (for short: tree to tree series transformation or t-ts transformation), and a mapping which transforms tree series into tree series (for short: tree series transformation or ts-ts transformation).

We show that the standard case of tree transducers is reobtained by choosing the boolean semiring under the t-ts semantics. Moreover, we show that certain fundamental constructions and results concerning bottom-up and top-down tree transducers can be generalized for the corresponding tree series transducers. Among others, we prove that polynomial bottom-up t-ts transformations can be characterized by the composition of finite state relabelling t-ts transformations and boolean homomorphism t-ts transformations. Moreover, we prove that every deterministic top-down t-ts transformation can be characterized by the composition of a boolean homomorphism t-ts transformation and a deterministic linear top-down t-ts transformation. We prove that deterministic top-down t-ts transformations are closed under right composition with nondeleting and linear deterministic top-down t-ts transformations and are closed under left composition with boolean and total deterministic top-down t-ts transformations. Finally we show that nondeleting linear bottom-up and nondeleting linear top-down tree series transducers generate the same t-ts transformation class.

In [8] we consider tree series transducers which were introduced in [6], and define the tree-to-tree series transformations computed by them in two different ways. One of the definitions is based on the $\longleftarrow$-substitution of tree series taken from [6] while the other one is based on a new tree series substitution introduced in this paper. This new substitution is called $\overset{o}{\longleftarrow}$-substitution and the main difference between the $\longleftarrow$- and the $\overset{o}{\longleftarrow}$-substitutions is that the first one does not take into account the number of the occurrences of the substitution variables while the second one does. We compare the two different ways of computing tree-to-tree series transformations and show that, for the $\overset{o}{\longleftarrow}$-substitution, fundamental relations from the theory of tree transducers carry over to tree series transducers.

In [9] we study bottom-up and top-down tree series transducers over a semiring A and denote the tree series transformation classes computed by them by $BOT_{t-ts}(A)$ and $TOP_{t-ts}(A)$, respectively. We present the inclusion diagram of the classes $p\text{-}BOT^n_{t-ts}(A)$, $p\text{-}TOP^n_{t-ts}(A)$, $p\text{-}BOT^{n+1}_{t-ts}(A)$, and $p\text{-}TOP^{n+1}_{t-ts}(A)$ and prove its correctness, where A is a commutative izz-semiring (izz = idempotent, zero-divisor free, and zero-sum free) and the prefix p stands for polynomial. This inclusion diagram implies the properness of the following four hierarchies:

$$p\text{-}TOP_{t-ts}(A) \subseteq p\text{-}TOP^2_{t-ts}(A) \subseteq p\text{-}TOP^3_{t-ts}(A) \subseteq \dots$$

$$p\text{-}BOT_{t-ts}(A) \subseteq p\text{-}BOT^2_{t-ts}(A) \subseteq p\text{-}BOT^3_{t-ts}(A) \subseteq \dots$$

$$p\text{-}TOP_{t-ts}(A) \subseteq p\text{-}BOT^2_{t-ts}(A) \subseteq p\text{-}TOP^3_{t-ts}(A) \subseteq p\text{-}BOT^4_{t-ts}(A) \subseteq \dots$$

$$p\text{-}BOT_{t-ts}(A) \subseteq p\text{-}TOP^2_{t-ts}(A) \subseteq p\text{-}BOT^3_{t-ts}(A) \subseteq p\text{-}TOP^4_{t-ts}(A) \subseteq \dots,$$

where the first hierarchy generalizes the famous top-down tree transformation hierarchy of J. Engelfriet. As the second main result we prove that the first two hierarchies are proper even for arbitrary (i.e., not necessarily commutative) izz-semirings.

In [6, 8] the semantics of tree series transducers was defined in an algebraic framework, more precisely, as an initial algebra semantics. In [10] we suggest an alternative approach by introducing weighted tree transducers, of which the semantics is defined in an operational style. A weighted tree transducer is a tree transducer each (term rewriting) rule of which is associated with a weight taken from a semiring. Along a successful derivation the weights of the involved rules are multiplied and, for every pair of input tree and output tree, the weights of its successful derivations are summed up. We show in a constructive way that the two approaches, i.e., tree series transducers and weighted tree transducers, are semantically equivalent for both, the top-down and the bottom-up case.

III. TREE AUTOMATA AND GROUND TERM REWRITING

In [4] we show that the component hierarchy of chain-free distributed regular tree grammars cooperating with terminal strategy is infinite with respect to tree language generating capacity. More exactly, we prove that $cfCDRTG_{\dashv}(n) \subset cfCDRTG_{\dashv}(2(n-1)^2+3)$, where $n \geq 1$ and $cfCDRTG_{\dashv}(n)$ denotes the class of tree languages generated by chain-free distributed regular tree grammars of at most n components cooperating with terminal strategy.

In [2] we consider restricted versions of ground tree transducers: total, deterministic, and symmetric subclasses and all other subclasses created by applying any combination of these restrictions. We present the inclusion diagram of the tree transformation classes induced by these restricted ground tree transducers.

We show that the following four classes of term relations are the same: (i) tree transformations induced by symmetric deterministic ground tree transducers, (ii) congruence relations on term algebras induced by reduced ground term rewriting systems, (iii) congruence relations on term algebras induced by convergent ground term rewriting systems, and (iv) finitely generated congruence relations on term algebras.

As a by-product of our results, we obtain a new ground completion algorithm.

Moreover, we show that the following three classes of term relations on term algebras with at least one non-nullary function symbol are also the same: (i) tree transformations induced by total symmetric deterministic ground tree transducers, (ii) congruence relations on term algebras of finite index, (iii) finitely generated congruence relations on term algebras of which the trunk is the whole set of terms.

CONFERENCE ORGANIZATION AND EDITORIAL WORK

In the period 2000-2003 Z. Fülöp served as the member of the Program Committee of the conferences Mathematical Foundations of Computer Science 2000, European Conference on Intelligent Systems and Technologies 2002, Developments in Language Theory 2002 and 2003 (co-chair), Automata and Formal Languages 2002. He was the coeditor of the proceedings [11] of DLT 03. He has been the editor of the journals Acta Cybernetica, Informatica, and Grammas.

ACKNOWLEDGEMENT

The above research has been supported by several grants from the National Foundation of Hungary for Scientific Research, the Exchange Program of the DAAD and the Hungarian Scholarship Committee and the International Quality Network program of the DAAD.

REFERENCES

[1] Z. Fülöp and S. Maneth, Domains of Partial Attributed Tree Transducers (S. Manethval), *Inf. Process. Letters*, **73** (2000) 175-180.

[2] Z. Fülöp and S. Vágvölgyi, Restricted Ground Tree Transducers, *Theoret. Comput. Science*, **250** (2001) 219-233.

[3] Z. Fülöp, Characterizing Important Subclasses of Attributed Tree Transformations by Macro Tree Transducers, *Journal of Automata, Languages and Combinatorics* **6** (2001) 145-189.

[4] G. Dányi and Z. Fülöp, The Component Hierarchy of Chain-Free Cooperating Distributed Regular Tree Grammar Systems, *Theoret. Comput. Science*, **262** (2001) 229-240.

[5] Z. Fülöp and A. Terlutte, Iterated Relabeling Tree Transducers, *Theoret. Comput. Science*, 276 (2002) 221-244.

[6] J. Engelfriet, Z. Fülöp and H. Vogler, Bottom-up and Top-down Tree Series Transformations,*Journal of Automata, Languages and Combinatorics*, **7** (2002) 11-70.

[7] Z. Fülöp and Zs. Gazdag, Shape Preserving Top-Down Tree Transducers, Theoret. Comput. Science, **304** (2003) 315-339.

[8] Z. Fülöp and H. Vogler, Tree Series Transformations that Respect Copying, Theory of Computing Systems, **36** (2003) 247-293.

[9] Z. Fülöp, Zs. Gazdag and H. Vogler, Hierarchies of Tree Series Transformations, *Theoret. Comput. Science*, accepted for publication.

[10] Z. Fülöp and H. Vogler, Weighted Tree Transducers (with H. Vogler), *Journal of Automata, Languages and Combinatorics*, accepted for publication.

[11] Proceedings of DLT 03 (editor Z. Ésik and Z. Fülöp), Lecture Notes in Computer Science 2710, Springer-Verlag, 2003.

STRUCTURE THEORY OF AUTOMATA

Ferenc Gécseg and Balázs Imreh

Structure theory of automata has been the source of many deep results. Properties of the members of special classes of automata and nondeterministic automata render it possible to

(1) describe the subdirect irreducible members of the class considered,

(2) help to characterize isomorphically and homomorphically complete systems for specific classes of automata under different compositions,

(3) describe the languages accepted by the members of the classes under consideration,

(4) study some questions related to directable automata.

As far as isomorphically complete systems are concerned, in [5] the class of the generalized definite automata is studied and the isomorphically complete systems of generalized definite automata with respect to the α_i-products are characterized. From this characterization it follows that there is no finite isomorphically complete system for this class with respect to the α_0-product, while there is a singleton isomorphically complete system for the class under consideration with respect to any α_i-product with $i \geq 1$. Asynchronous automata are studied in [11] and the isomorphically complete systems are characterized for this class with respect to the α_i-products. Isomorphically complete systems for the class of all automata under different compositions are also studied. In [10] it is shown that the cube-product is equivalent to the Glushkov's product of automata, and thus, the isomorphically complete systems collapse for these two compositions.

More general classes, the class of monotone tree automata, and the class of nondeterministic monotone tree automata are investigated in [4] where the isomorphically complete systems are characterized for these classes with respect to the α_0-products. From these descriptions it follows that there are singleton isomorphically complete systems for both classes with respect to the α_0-product. The class of commutative asynchronous nondeterministic automata is studied in [15] with respect to the isomorphically complete systems. In [1] a structural characterization of three classes of tree automata is presented, namely, the classes of nilpotent, definite, and monotone tree automata are homomorphically represented by means of quasi-cascade-products of unary nilpotent and unary definite tree automata in the first two cases, and by means of products of simpler tree automata in the third case. Another specific class, the class of asynchronous tree automata is studied in [9] where the isomorphically complete systems are described for this class with respect to the α_i-products.

It is known that the products of tree automatado not preserve the basic properties of homomorphically and metrically complete systems of finite state automata. To remedy it, a new concept, the quasi-product is introduced in [2] which is a slightly more general than the product. The main properties of the quasi-product concerning homomorphic and metric representations of tree automata are studied in [2] and [3].

Different special classes of languages are described in the papers [7, 6, 8, 9, 11, 14, 17]. In [17] the classes of languages consisting of the languages containing the directing words of commutative nondeterministic directable automata are described and compared. A small particular language class, the class of the languages accepted by commutative asynchronous automata is described in [14]. Another special class, the class of the languages recognizable by monotone automata is characterized in [6] and the languages recognized by asynchronous automata are described in [11].

The tree languages recognized by deterministic root-to-frontier tree automata are characterized in [7]. The concepts of definite and nilpotent DR tree languages are introduced in [8] and they are characterized by means of syntactic path semigroups. The tree languages recognized by asynchronous tree automata are described in [9].

In [16] the directable commutative asynchronous automata are studied and a tight upper bound is presented for the maximum of the shortest directing words of n-state directable commutative asynchronous automata.

The notion of the directability of automata is extended to nondeterministic automata in [18] where lower and upper bounds of the maximum of the shortest directing words of n-state directable nondeterministic automata are investigated. This extension yielded further studies for commutative nondeterministic automata in [17], for nondeterministic trapped automata in [12], and for monotone nondeterministic automata in [13].

RELATIONSHIP BETWEEN THIS RESEARCH TOPIC AND THE EDUCATION

Results concerning the structure studies of automata can constitute different PhD courses, furthermore, they provide numerous possibilities for further research work.

ACKNOWLEDGEMENT

Research sketched above was supported by the Hungarian National Foundation for Scientific Research, Grants T 030143 and T 037258, the Japanese Ministry of Education, Mombusho International Scientific Research Program, Joint Research 10044098.

REFERENCES

[1] Gécseg, F., On Some Classes of Tree Automata and Tree Languages, *Annales Academiae Scientiarum Fennicae Mathematica* **25** (2000), 325-336.

[2] Gécseg, F., On quasi-products of tree automata, *Journal of Universal Computer Science* **2** (2002), 184-192.

[3] Gécseg, F., Comments on complete sets of tree automata, in: Proceedings of 7th International Conference, Developments in Language Theory, eds.: Z. Ésik, Z. Fülöp, 2003. Szeged, Springer-Verlag, LNCS **2710** (2003), 22-34.

[4] Gécseg, F., B. Imreh, On isomorphic representation of monotone tree and nondeterministic tree automata, in: *Words, Semigroups, Transductions*, eds.: M. Ito, Gh. Paun, S. Yu, World Scientific, New Yersey – London – Singapore – Hong Kong, 2001, 141-154.

[5] Gécseg, F., B. Imreh, On isomorphic representations of generalized definite automata, *Acta Cybernetica* **15** (2001), 33-44.

[6] Gécseg, F., B. Imreh, On monotone automata and monotone languages, *Journal of Automata, Languages and Combinatorics*, **7** (2002), 71-82.

[7] Gécseg, F. and Steinby, M.: Minimal recognizers and syntactic monoids of DR tree languages, in: *Words, Semigroups, Transductions*, eds.: M. Ito, Gh. Paun, S. Yu, World Scientific, New Yersey – London – Singapore – Hong Kong, 2001, 156-167.

[8] Gécseg, F., B. Imreh, On definite and nilpotent DR tree languages, *Journal of Automata, Languages and Combinatorics*, to appear.

[9] Gécseg, F., B. Imreh, On asynchronous tree automata, *Journal of Automata, Languages and Combinatorics*, submitted for publication

[10] Imreh, B., On the equivalence of the cube-product and the generalized product of automata, *Theoretical Computer Science* **245** (2000), 103-113.

[11] Imreh, B., Some Remarks on Asynchronous Automata, in: Proceedings of 6th International Conference, Developments in Language Theory, eds.: M. Ito, M. Toyama, 2002. Kyoto, Springer-Verlag, LNCS **2450** (2003), 290–296.

[12] Imreh, B., Cs. Imreh, M. Ito, On directable nondeterministic trapped automata, *Acta Cybernetica*, **16** (2003), 37-45.

[13] Imreh, B., Cs. Imreh, M. Ito, On directable nondeterministic monotonic automata, *Journal of Automata, Languages and Combinatorics*, **8** (2003), 539–547.

[14] Imreh, B., M. Ito, A. Pukler, A note on the languages recognized by commutative asynchronous automata, *RIMS Kokyuroku* **1166** (2000), 95-99.

[15] Imreh, B., M. Ito, A. Pukler, On commutative asynchronous nondeterministic automata, *Acta Cybernetica* **14** (2000), 607-617.

[16] Imreh, B., M. Ito, A. Pukler, On commutative asynchronous automata, in: Proceedings of the Third International Colloquium on Words, Languages, and Combinatorics, Kyoto, 2000, to appear.

[17] Imreh, B., M. Ito, M. Steinby, On commutative directable nondeterministic automata, in: *Grammars and Automata for String Processing from Mathematics and Computer Science to Biology and Back*, eds.: Carlos Martin-Vide, Victor Mitrana, Taylor and Francis, (Topics in Computer Mathematics), London, 2002, 141-150.

[18] Imreh, B., M. Steinby, Directable nondeterministic automata, *Acta Cybernetica* **14** (1999), 105-115.

STOCHASTIC MODELING IN IMAGE PROCESSING

Zoltan Kato

MRF (Markov Random Field) modeling and MCMC (Markov Chain Monte Carlo) methods are successfully used in different areas of image processing [3]. Herein, we will consider two topics: The first one deals with multicue MRF modeling applied to the segmentation of color textured images [1, 4]. The second one is a novel image similarity measure based on a stochastic painting process, which is applied to content based image retrieval [2, 5].

I. MULTI-LAYER MRF MODELING

We propose a new MRF image segmentation model which aims at combining color and texture features. The model has a multi-layer structure: Each feature has its own layer, called *feature layer*, where an MRF model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model. This layer interacts with each feature layer and provides the segmentation based on the combination of different features. The uniqueness of our algorithm is that it provides both color only and texture only segmentations as well as a segmentation based on combined color *and* texture features. The number of classes on feature layers is given by the user but it is estimated on the combined layer.

Our model consists of 3 layers. At each layer, we use a first order neighborhood system and extra inter-layer cliques (Fig. 1). The image features are represented by multi-variate Gaussian distributions. Let us denote the color layer by $\mathcal{S}^c$, the texture layer by $\mathcal{S}^t$ and the combined layer by $\mathcal{S}^x$. All layers are of the same size. Our MRF model is defined over the lattice $\mathcal{S} = \mathcal{S}^c \cup \mathcal{S}^x \cup \mathcal{S}^t$. For each site s, the region-type (or class) that the site belongs to is specified by a class label, ω_s, which is modeled as a discrete random variable taking values in $\Lambda = \{1, 2, \ldots, L\}$. The set of these labels $\omega = \{\omega_s, s \in \mathcal{S}\}$ is a random field, called the *label process*. Furthermore, the observed image features (color and texture) are supposed to be a realization $\mathcal{F} = \{\vec{\mathbf{f}}_s | s \in \mathcal{S}^{\rfloor} \cup \mathcal{S}^{\sqcup}\}$ from another random field, which is a function of the label process ω. Basically, the *image process* $\mathcal{F}$ represents the deviation from the underlying label process. Thus, the overall segmentation model is composed of the hidden label process ω and the observable noisy image process $\mathcal{F}$. Our goal is to find an optimal labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega \mid \mathcal{F})$, that is the *maximum a posteriori* (MAP) estimate: $\arg\max_{\omega\in\Omega} P(\omega \mid \mathcal{F}) = \arg\max_{\omega\in\Omega} \prod_{s\in\mathcal{S}} P(\vec{\mathbf{f}}_s \mid \omega_s)P(\omega)$, where Ω denotes the set of all possible labelings. We use the ICM algorithm to obtain a suboptimal MAP estimate. According to the *Hammersley-Clifford theorem*, $P(\omega \mid \mathcal{F})$ follows a Gibbs distribution:

$$P(\omega \mid \mathcal{F}) = \frac{\exp(-U(\omega))}{Z(\beta)} = \frac{\prod_{C\in\mathcal{C}} \exp(-V_C(\omega_C))}{Z(\beta)} \tag{2}$$

where $U(\omega)$ is called an *energy function*, $Z(\beta) = \sum_{\omega\in\Omega} \exp(-U(\omega))$ is the normalizing constant and V_C denotes the *clique potential* of clique $C \in \mathcal{C}$ having the label configuration ω_C. Note that the energies of *singletons* (ie. cliques of single sites $s \in \mathcal{S}$) directly reflect the probabilistic modeling of labels without context, while higher order clique potentials express relationship between neighboring pixel labels.

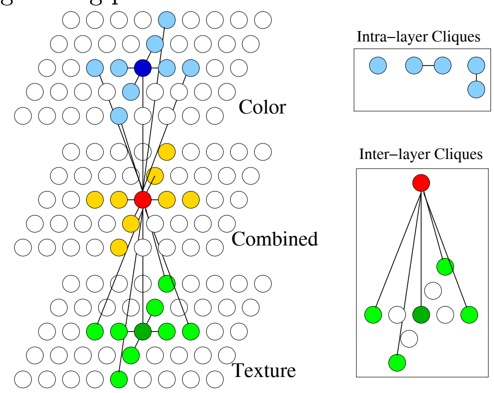

Figure 1. Multi-layer MRF model.

On the color layer, the observed image $\mathcal{F}^c = \{\vec{\mathbf{f}}^c_s | s \in \mathcal{S}^c\}$ consists of three spectral component values ($L^*u^*v^*$) at each pixel s denoted by the vector $\vec{\mathbf{f}}^c_s$. We assume that $P(\vec{\mathbf{f}}^c_s \mid \omega_s)$ follows a Gaussian distribution, the classes $\lambda \in \Lambda^c = \{1, 2, \ldots, L^c\}$ are represented by the mean vectors $\vec{\mu^{\mathbf{c}}}_\lambda$ and the covariance matrices $\mathbf{\Sigma^c}_\lambda$. The class label assigned to a site s on the color layer is denoted by ψ_s. The energy function $U(\psi, \mathcal{F}^c)$ of the so defined MRF layer has the following form:

$$\sum_{s\in\mathcal{S}^c} \mathcal{G}^c(\vec{\mathbf{f}}^c_s, \psi_s) + \beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\psi_s, \psi_r) + \sum_{s\in\mathcal{S}^c} V^c(\psi_s, \eta^c_s)$$

where $\mathcal{G}^c(\vec{\mathbf{f}}^c_s, \psi_s)$ denotes the Gaussian energy term while $\delta(\psi_s, \psi_r) = 1$ if ψ_s and ψ_r are different and -1 otherwise. $\beta > 0$ is a parameter controlling the homogeneity of the regions. As β increases, the resulting regions become more homogeneous. The last term ($V^c(\psi_s, \eta^c_s)$) is the inter-layer clique potential.

On the texture layer, the observation consists of a set of Gabor image features. To obtain these features, we use a multi-channel filtering approach. The channels are represented by a bank of real-valued, even-symmetric Gabor filters. The MRF model itself is quite similar to the color layer with obvious modifications.

The combined layer only uses the texture and color features indirectly, through inter-layer cliques. A label consists of a pair of color and texture labels such that $\eta = \langle \eta^c, \eta^t \rangle$, where $\eta^c \in \Lambda^c$ and $\eta^t \in \Lambda^t$. The set of labels is denoted by $\Lambda^x = \Lambda^c \times \Lambda^t$ and the number of classes $L^x = L^c L^t$. Obviously, not all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chose those pairs of texture and color labels which are actually present in a given image. The energy function $U(\eta)$ is of the following form:

$$\sum_{s \in \mathcal{S}^x} \left(V_s(\eta_s) + V^c(\psi_s, \eta_s^c) + V^t(\phi_s, \eta_s^t) \right) + \alpha \sum_{\{s,r\} \in \mathcal{C}} \delta(\eta_s, \eta_r)$$

where $V_s(\eta_s)$ denotes singleton energies, $V^c(\psi_s, \eta_s^c)$ (resp. $V^t(\phi_s, \eta_s^t)$ denotes inter-layer clique potentials. The last term corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s, \eta_r) = -1$ if $\eta_s = \eta_r$, 0 if $\eta_s \neq \eta_r$ and 1 if $\eta_s^c = \eta_r^c$ *and* $\eta_s^t \neq \eta_r^t$ or $\eta_s^c \neq \eta_r^c$ *and* $\eta_s^t = \eta_r^t$. The idea is that region boundaries present at both color and texture layers are preferred over edges that are found only at one of the feature layers. Inter-layer interactions are as follows:

$$V^c(\psi_s, \eta_s^c) = \gamma^c \sum_{\{s,r\} \in \mathcal{C}_5} W_r D^c(\psi_r, \eta_s^c)$$

where $D^c(\psi_r, \eta_s^c) = \mid \mathcal{G}^c(\vec{\mathbf{f}}_r^c, \psi_r) - \mathcal{G}^c(\vec{\mathbf{f}}_s^c, \eta_s^c) \mid$. $V^t(\phi_s, \eta_s^t)$ and $D^t(\phi_r, \eta_s^t)$ are defined in a similar way using texture features and γ^t as weight. At any site s, we have 5 inter-layer interactions (denoted by $\mathcal{C}_5$) between two layers: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Fig. 1). W_r is the weight of the clique $\{s,r\} \in \mathcal{C}_5$. We assign higher weight (0.6) to the corresponding site whereas smaller weights (0.1 each) to the other 4 neighboring sites. The latter 4 sites help to ensure homogeneity on the combined layer. Note that D^c and D^t corresponds to the difference of the first order potentials at the corresponding feature layer. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. If the labels are different then it is proportional to the energy difference between the two labels. γ^c (resp. γ^t) controls the influence of the inter-layer cliques. A higher value will increase the importance of the information coming from the other layer.

The singleton energy is defined as follows:

$$V_s(\eta_s) = R((10N_{\eta_s})^{-3} + \mathcal{P}(L)) + \rho^c \sum_{\{s,r\} \in \mathcal{C}_5} W_r D^c(\psi_r, \eta_s^c) + \rho^t \sum_{\{s,r\} \in \mathcal{C}_5} W_r D^t(\phi_r, \eta_s^t)$$

The importance of the above potential function is twofold: First of all, it controls the number of classes at the combined layer. $(10N_{\eta_s})^{-3}$ penalizes small classes (N_{η_s} is the percentage of the sites assigned to class η_s), while $\mathcal{P}(L)$ includes some prior knowledge about the number of classes. Currently this is expressed by a log Gaussian term with mean value $\hat{L}$ (basically an initial guess) and variance σ (confidence in the initial guess). R is simply a weight of this term, we set it to 0.5 in our tests. Second, it balances the influence of the feature layers to the combined layer. This is achieved by including D^c and D^t in the potential function. ρ^t (resp. ρ^c) controls this balancing: When the clique potentials are summed for a given site *on the combined layer*, the final weight of D^t equals to $\rho^t + \gamma^t$. However, the weight of D^t at a site *on the texture layer* is always γ^t. Therefore, depending on the value of ρ^t (resp. ρ^c), we can increase or decrease the influence of a feature layer to the combined layer without changing the influence of the combined layer to a feature layer. We found this an important issue in the case of the texture layer. Fig. 2 shows some segmentation results together with the measured misclassification rate. Clearly, the multi-layer model provides significantly better results compared to color only and texture only segmentations. See also (`www.inf.u-szeged.hu/~kato/reserach/icpr2002/`)

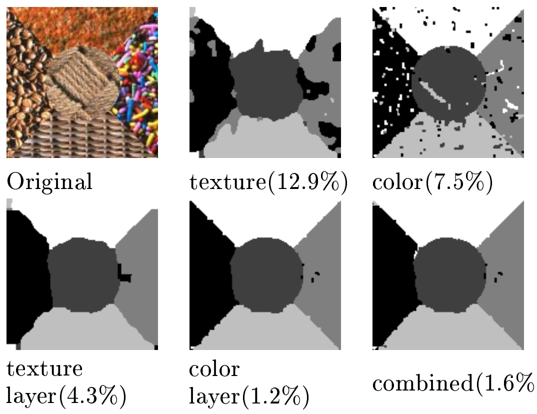

Figure 2. Results and misclassification rate of color only, texture only, and combined models

II. CONTENT BASED IMAGE RETRIEVAL

Content-based image retrieval (CBIR) deals with the indexing and retrieval of images in a large database based on some low level visual features (so called visual content). The basic question when dealing with the visual content of an image is how humans can interpret an image. In the biological sense, it is not an easy question. However, there is an offering answer from an artistic point of view: ask a talented painter and he will give a painted interpretation of the world: the scene as the artist sees it. Such an image is made of brush-strokes of different sizes and colors put on the screen one after the other in a sequence by the painter. Small details are elaborated with fine brushes, while plain surfaces are painted with greater strokes.

Stochastic Paintbrush Transformation (SPT) is a method to simulate such a painting process. Brush strokes are generated randomly at decreasing scales of brush-sizes. The strokes are then either accepted or rejected based on the change in distortion they introduce. Basically, it can be regarded as a multi-scale image decomposition method, based on simulated, arbitrarily shaped paintbrush strokes. A stroke is determined by five parameters: shape, size, location, orientation and color. Herein, we use elliptical-shaped brushes because it provides better edges. Brush size is decreased at every painting stage. Color is determined by the majority color in the stroke area of the original image while location and orientation are randomly generated. It is also important that we use the perceptually uniform CIE-L*u*v* color space. A new stroke is accepted according to a probabilistic rule which prefers strokes decreasing the distortion of the painting. The resulting images look like good-quality paintings with well-defined contours, at an acceptable distortion compared to the original image (See Fig. 3). Basically, the painting algorithm is formulated as an optimization problem which is solved by simulated annealing (SA).

The next question is how to measure the similarity of a pair of images using stroke parameters. In fact, the painting process provides a semi-segmented image: important edges are preserved and there are no fine details below a limit (determined by the brush size). However, unlike real segmentation, objects are usually covered by several overlapping strokes. The basic idea here is that similar images have similar painted representations. Hence, our measure is based on matching brush-strokes using their size, color, orientation and location in the query image Q and a candidate image R from the database. The matching is done level by level (one level consists of strokes with the same brush size) based on the color, orientation and location of a pair of strokes.

We start with matching the color parameter because it is the most important feature. Then we match the orientation which characterizes the structure of the image. Finally the location of the strokes are matched which tells us about the spatial location of a stroke. The algorithm produces l lists, each of them having n_l matching pairs of brush strokes: $\{\langle \mathcal{B}^1_{Q_l}, \mathcal{B}^1_{R_l}\rangle, \ldots, \{\langle \mathcal{B}^n_{Q_l}, \mathcal{B}^{n_l}_{R_l}\rangle\}$. Besides having the same shape and size, we expect that each pair have similar colors, same orientation and they are close to each other. Note that we do not require an exact matching and not all strokes have a matching pair. The similarity $\mathcal{S}_l$ of two images at a given brush size l is defined as

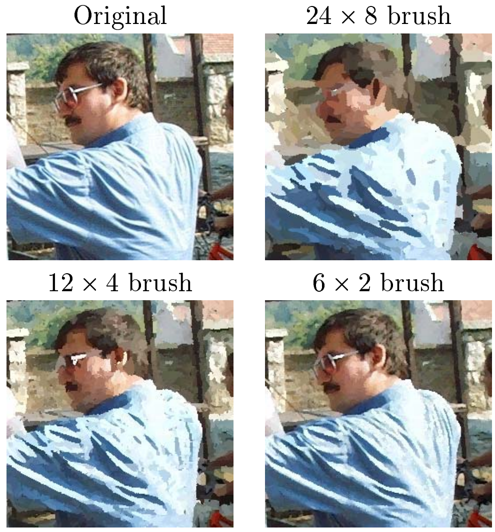

Figure 3. Typical paintings produced by SPT from a real color image with different brush sizes.

$$\mathcal{S}_l = \frac{1}{N_l}\sum_{i=1}^{n_l} 1 - \frac{\delta(\mathcal{B}^i_{Q_l}, \mathcal{B}^i_{R_l})}{D} \tag{3}$$

where $\delta()$ denotes the distance between the location of two strokes, N_l is the number of strokes in Q_l, and D is the largest distance in the query image. Therefore the value $\mathcal{S}_l \in [0, 1]$ depends on the number of matching pairs *and* the distance between the pairs. Using Eq. (3), we define the similarity of a pair of images as the normalized sum of the similarity values weighted by the coverage area:

$$\mathcal{S} = \frac{1}{\mathcal{A}}\sum_{i=1}^{n} \mathcal{S}_i \mathcal{A}_i \tag{4}$$

where n is the number of different brush sizes used by SPT, $\mathcal{A}$ is the area of Q, and $\mathcal{A}_l$ is the coverage area of the strokes in Q_l.

For the evaluation of the proposed CBIR method, we have used a ground truth produced by manual classification of images. The whole database is available on our web site (`www.inf.u-szeged.hu/~kato/reserach/spt/cbir/`). Experiments show (see Fig. 4) that similarity based on stroke parameters has higher retrieval rate than traditional color-based similarity or Oracle 9i's CBIR function.

III. CONNECTIONS OF RESEARCH TO EDUCATION

These research topics are closely related to the *Digital Image Segmentation* and the *Energy Minimization Methods in Image Segmentation* special courses as well as to the *Markov Random Fields in Image Processing* PhD course. There are also ongoing Masters thesis works related to these topics. At the time of writing this report, one of our students, Peter Horvath won the first price of the local Student Research Competition (TDK) with

Figure 4. Retrieval results of the proposed method, histogram intersection, and Oracle's CBIR function.

his project entitled "Motion Detection and Tracking Using a Robot Camera".

IV. CONFERENCE ORGANIZATION, EDITORIAL WORK

I've participated in the organization of *IEEE International Conference on Image Processing* 2004, one of the major events of the image processing community, as a Technical Program Committee member. Since 2003, I'm an Associate Editor of *IEEE Transactions on Image Processing*, responsible for the area *"Segmentation"*. I have also refereed a couple of papers submitted to IEEE-ACCV 2002, ICPR 2002, WOMTEC 2003, IEEE Trans. on Image Processing, IEEE Trans. on Medical Imaging, IEEE Signal Processing Letters, Image and Vision Computing, Pattern Recognition Letters.

V. COLLABORATIONS

Zoltan Toth and Dr. Tamas Sziranyi from Image Processing and Neurocomputing Department of Veszprem University [2]. Ji Xiaowen [2, 5], Dr. Huang Zhiyong [5] and Song Guo Qiang [1, 4] from School of Computing, National University of Singapore. Dr. Ting Chuen Pong from Computer Science Department of Hong Kong University of Science and Technology [1, 4]. Andre Jalobeanu and Dr. Josiane Zerubia from ARIANA project, INRIA-Sophia Antipolis, France [3].

ACKNOWLEDGEMENTS

Our research has been supported by the following grants: National University of Singapore Grant R-252-000-085-112 (Principal Investigator), Bolyai Janos Research Fellowship of the Hungarian Academy of Sciences.

REFERENCES

[1] Z. Kato, T. C. Pong, and S. G. Qiang, "Multicue mrf image segmentation: Combining texture and color", in *Proceedings of the International Conference on Pattern Recognition*, (Quebec, Canada), IEEE-IAPR, Aug. 2002.

[2] Z. Kato, Ji Xiaowen, T. Sziranyi, Z. Toth, and L. Czuni, "Content-Based Image Retrieval Using Stochastic Paintbrush Transformation", in *Proceedings of International Conference on Image Processing*, (New York, USA), IEEE, Sep. 2002.

[3] J. Zerubia, A. Jalobeanu, and Z. Kato, "Markov random fields in image processing, application to remote sensing and astrophysics", *Journal de Physique, EDP Sciences*, vol. IV, no. 12, 2002.

[4] Z. Kato, T. C. Pong, and S. G. Qiang, "Unsupervised segmentation of color textured images using a multi-layer MRF model", in *Proceedings of International Conference on Image Processing*, vol. I, (Barcelona, Spain), pp. 961–964, IEEE, Sep. 2003.

[5] Ji Xiaowen, Z. Kato, and Huang Zhiyong, "Non-Photorealistic Rendering and Content-Based Image Retrieval", in *Proceedings of Pacific Conference on Computer Graphics and Applications*, (Canmore, Canada), pp. 153, IEEE Oct. 2003.

MACHINE LEARNING AND ITS APPLICATION TO SPEECH TECHNOLOGY

András Kocsor, Kornél Kovács, and László Tóth

I. MOTIVATION

Our research focuses on machine learning, the most intensively studied application field here being speech recognition. This application seemed interesting because speech recognition has its unique solutions and the latest methods of machine learning find their way into it very slowly. To have a proper environment for experimentation, we looked for a flexible framework that allows the combination of different preprocessing techniques, feature-space transformation methods and machine learning algorithms. These expectations led us to adopt the stochastic segmental approach which may be viewed as an extension of hidden Markov modeling. In this framework we performed numerous studies on the applicability of several machine learning algorithms and transformation methods to the speech recognition task. Although speech recognition is the main line of our research, we also work on some related areas like speech synthesis and the application of our machine learning algorithms in other fields.

II. MACHINE LEARNING ALGORITHMS

We have implemented several well-known and not so well-known machine learning algorithms like Artificial Neural Nets, Gaussian Mixture Models, Support Vector Machines, Projection Pursuit Learning. Moreover, within the framework of an SZT-IS-10 grant we made the code for these algorithms publicly available via our website. The performance of these algorithms was compared with speech recognition (phoneme classification) tasks [3, 24].

III. LINEAR AND NON-LINEAR FEATURE SPACE TRANSFORMATIONS

These classifiers can be made even more efficient with the aid of feature-space transformations. There are some traditional linear transformation methods for this goal, but making them non-linear by the kernel method has become a very popular research trend recently. One of our main research interests currently is the application of non-linear transformations to phoneme classification. We carried out several studies that investigated the efficiency of transformations like (Kernel-)LDA, (Kernel-)PCA, (Kernel-)ICA and (Kernel-)SDA [5, 6, 7, 9, 25].

IV. THE "OASIS" RECOGNITION SYSTEM

Our speech recognition system follows the stochastic segmental approach and was named "OASIS" [1, 2, 4] (from Our Acoustics-based Speaker-Independent Speech recognizer). The system was designed to be as modular as possible, so we can easily conduct experiments by combining different techniques for the several subtasks of recognition. Currently the system consists of the following modules:

IV.1. THE PHONEME CLASSIFIER

The very first step of recognition is the usual frame-based preprocessing. After this, however, the segmental approach has an additional step, which represents the segments as one unit. This allows for the combination of different preprocessing techniques and also the incorporation of phonetic knowledge. The output of this step is the so-called segmental feature set, which serves as input for the phoneme classifier. The main advantage of the segmental approach from our point of view is that one can apply many traditional pattern recognition algorithms for the phoneme classification task. This is the most intensively studied component of our system so far, and we reported in several papers that discriminative classifiers like Artificial Neural Nets or Support Vector Machines can significantly outperform the usual HMM technique even with a very simple segmental feature set. [3, 24].

IV.2. THE ANTI-PHONE COMPONENT

One drawback of the segmental representation is that the phoneme classifier is not automatically able to handle those segments that do not correspond to real phones. However, during the search process the matching engine faces such hypothesized segments and has to recognize them. In our system this problem is solved by an additional component, the anti-phone model, whose precise task is to handle these segments during the search. In [4] we described how these segments are modelled, trained and combined in our system. Recently an alternative solution was also proposed that uses Replicator Neural Networks [26].

IV.3. THE LANGUAGE MODEL

In practical recognition tasks there is a serious restriction on the possible words accepted by the system. These words and their corresponding probabilities are represented by the language model. Up to this point we have used only very simple language models such as a dictionary of words and a unigram language model, but our system is theoretically capable of handling more complicated representations as well. The interface and the proposed working scheme of our language model was recently given in [23].

IV.4. THE MATCHING ENGINE

The information of the knowledge sources (phoneme classifier, anti-phone component, language model) are combined by the matching engine. This performs an utterance-level search in the graph of the possible segmentations. Its goal is to find the best matching transcription from those provided by the language model. The system allows one to experiment with many search techniques and strategies. We studied several pruning and search algorithms for making the matching engine more efficient [19, 10].

V. RUNNING PROJECTS AND COMPLETED ONES

A very important issue in statistical speech recognition is the training database. Without a proper corpus the recognition system cannot be trained reliably. Unfortunately, no such corpora for Hungarian existed that was large enough to train large vocabulary continuous speech recognizers. In the framework of the 'IKTA-3' national grants we collected a large telephone-based speech corpus, working in cooperation with the Technical University of Budapest [12, 22]. We have also made a small corpus containing natural numbers, and this latter corpus is made available to any research institute in Hungary, within the framework of project funded by an SZT-IS-10 grant. This project also included the creation of free source-code algorithms that aim to encourage speech recognition research and development in Hungary.

Besides these, we have just finished a project in the field of speech impediment therapy that was sponsored by the IKTA-4 grant (for details see below). The resulting "BeszédMester" software is now being tested in many elementary schools, and the results so far are encouraging.

In addition, we have just started another project within the IKTA-6 framework that seeks to collect a large speech corpus. The structure of this corpus is similar to that of the telephone-based one, but this is recorded directly into the computer. This project also includes the creation of a speech recognizer for medical dictation tasks.

VI. SPEECH IMPEDIMENT THERAPY

People with hearing impairment face terrible difficulties when learning to speak. Because the auditory feedback is missing, their pronunciation is usually slurred and hard to understand. The missing feedback can be partially replaced by a visual one, by means of computer programs that represent speech in visual form. This helps them understand the connection between their own articulation and the speech signal they produce. The usual technology for speech visualization is to display some kind of (reduced) spectral representation. The main problem with this is that a very simple representation is required - otherwise the eye is unable to follow the signal in real time. This is especially true for young children, whom one cannot expect to understand rapidly changing spectral curves. Thus we apply speech recognition to this task, which permits feedback in the form of flickering letters. These are much easier to follow with the eye, and they also help teach reading, which, for hearing impaired children, usually runs in parallel. We have developed the system called "BeszédMester" with two applications in mind. One is the speech therapy of children with hearing difficulties. The other is to help children learn to read - this is for children with impaired or normal hearing. The software uses up-to-date machine learning technologies to classify speech sounds in real-time. The sound corpus for training the system is collected by two cooperative partners, the Kindergarten, Primary school and Boarding school of Kaposvár and by the University of Szeged Teacher Training School.

VII. DATABASES

The MTBA Hungarian Telephone Speech Database contains 500 phone calls, each containing the recordings of city names, numbers, company names, and so on, and, most importantly, 6000 phonetically balanced sentences that are manually segmented and labelled. Upon payment, this corpus is available for both research and industrial purposes.

The MRBA database is designed to be similar in structure, but it is recorded via microphones connected directly to the computer. This corpus should be completed and made available in 2006.

The Oasis-Number database contains the pronunciation of numbers and is freely available for any user. It can be downloaded from our home page.

Finally, during the creation of the "BeszédMester" software we collected a large amount of recordings from children. Although this corpus is not publicly available, we plan further research on this, and we are also open to any kind of research cooperation using this database.

VIII. SPEECH SYNTHESIS

Recently we started to develop a general-purpose text-to-speech system for Hungarian. It is based on a concatenative synthesis technique and our overall aim is to make it applicable to dialogue systems. Currently the system is still in its infancy as it cannot handle speech prosody as yet [13].

REFERENCES

[1] Kocsor, A., Kuba A. Jr., Tóth L., Jelasity M., Felföldi L., Gyimóthy T., Csirik J.,: A Segment-Based Statistical Speech Recognition System for Isolated/Continuous Number Recognition, Proceedings of the FUSST'99, Aug. 19-21, Sagadi,Estonia, pp. 201-211, 1999.

[2] Kocsor, A., Kuba A. Jr., Tóth L.: An Overview of the OASIS speech recognition project, Proceedings of the 4th International Conference on Applied Informatics, Aug. 30-Sept. 3, Eger-Noszvaj, Hungary, pp. 94-102, 1999.

[3] Kocsor, A., Tóth, L., Kuba, A. Jr., Kovács, K., Jelasity, M., Gyimothy, T., Csirik, J.: A Comparative Study of Several Feature Space Transformation and Learning Methods for Phoneme Classification, International Journal of Speech Technology, Vol. 3, Number 3/4, pp. 263-276, 2000.

[4] Tóth, L., Kocsor, A., Kovács, K.: A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition, Third International Workshop on Text, Speech and Dialogue, Brno, Czech Republic, 13-16 September, LNAI 1902, PP. 307-313, Spreinger Verlag, 2000.

[5] Kocsor, A., Kuba, A. Jr., Tóth, L.: Phoneme Classification Using Kernel Principal Component Analysis, Periodica Polytechnica, Vol. 44, No. 1, pp. 77-90, 2000.

[6] Kocsor, A., Tóth, L., Felföldi, L.: Application of Feature Transformation and Learning Methods in Phoneme Classification, in: L. Monostori, J. Váncza and M. Ali (Eds.): Proc. 14th Int. Conf. On Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2001, LNAI 2070, pp. 502-512, Springer Verlag, 2001.

[7] Kocsor, A., Tóth, L., Paczolay, D.: A Nonlinearized Discriminant Analysis and its Application to Speech Impediment Therapy, in: Proceedings of the 4th Int. Conf. On Text, Speech and Dialogue, LNAI 2166, pp. 249-257, Springer, 2001.

[8] Kocsor, A., Dombi, J., Bálint, I.: An Optimization Based Algorithm for Determining Eigenpairs of Large Real Matrices, Annales Univ. Sci. Budapest., Sect. Comp., Vol. 20, pp. 237-253, 2001.

[9] Kocsor, A., Csirik, J..: Fast Independent Component Analysis in Kernel Feature Spaces, in: L. Pacholski and P. Ruzicka (Eds.): Proceedings of SOFSEM 2001: Theory and Practice of Informatics: 28th Conference on Current Trends in Theory and Practice of Informatics, LNCS 2234, pp. 271-281, Springer Verlag, 2001.

[10] Felföldi, L., Kocsor, A., Tóth, L.: Classifier Combination in Speech Recognition, Periodica Polytechnica Ser. El. Eng., Vol. 41, No. 1., pp. 1-16, 2002.

[11] Kocsor, A., Kovács, K.: Kernel Springy Discriminant Analysis and Its Application to a Phonological Awareness Teaching System, in: P. Sojka, I. Kopecek, K. Pala (Eds.): Proceedings of Text, Speech and Dialogue 5th International Conference, TSD 2002, LNAI 2448, pp. 325-328, Springer Verlag, 2002.

[12] Vicsi, K., Tóth, L., Kocsor, A., Gordos, G., Csirik, J.: (in Hungarian) MTBA - magyar nyelvű telefonbeszéd-adatbázis, Hiradastechnika, Vol. LVII, NO.8, 2002

[13] Kovács, K., Kocsor, A., Tóth, L.: Hungarian Speech Synthesis Using a Phase Exact HNM Approach, in: Grosky, W. I., Plasil, F. (eds.): Proc. of SOFSEM 2002: Theory and Practice of Informatics, Milovy, Czech Republic, Nov. 2002, pp. 181-185

[14] Paczolay, D., Tóth, L., Kocsor, A., Kerekes, J.: (in Hungarian) Gépi tanulás alkalmazása egy fonológiai tudatosság-fejlesztő rendszerben, Alkalmazott Nyelvtudomány, II. évf. 2. szám, 2002, pp. 55-67

[15] Kocsor, A., Dombi, J., Bálint, I.: Inequality-Based Approximation of Matrix Eigenvectors, International Journal of Applied Mathematics and Computer Science, Vol. 12, No. 4, pp. 533-538, 2002.

[16] Paczolay. D., Kocsor, A., Tóth, L.: Real-Time Vocal Tract Length Normalization in a Phonological Awareness Teaching System, Matousek, V., Mautner, . (eds.): Proc. of TSD'2003, pp. 309-314.

[17] Kocsor, A, Paczolay, D.: (in Hungarian) Gépi tanulás
és alkalmazása a beszédjavítás-terápiában, XIII. Magyar Alkalmazott Nyelvészeti Kongresszus, MANYE, összefoglaló kötete, p. 91, 2003.

[18] Gosztolya, G., Kocsor, A.: Improving the Multistack Decoding Algorithm in a Segment-based Speech Recognizer, in: P. W. H. Chung, C. J. Hinde, M. Ali (Eds.): Proceedings of 16th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2003, LNAI 2718, pp. 744-749, Springer Verlag, 2003.

[19] Gosztolya, G., Kocsor, A., Tóth, L., Felföldi, L.: Various Robust Search Methods in a Hungarian Speech Recognition System, Acta Cybernetica, Vol. 16, No. 2., pp. 229-240, 2003.

[20] Kertész-Farkas, A., Fülöp, Z., Kocsor, A.: (in Hungarian) Magyar nyelvű szótárak tömör reprezentációja nemdeterminisztikus automatákkal, I. Magyar Számítógépes Nyelvészet Konferencia, Dec. 10-11., pp. 231-237, Szeged, 2003.

[21] Tóth, L., Kocsor, A., Harmonic Alternatives to Sine-Wave Speech, Proc. Eurospeech, Geneva, 2003, pp. 2073-2076.

[22] Tóth, L., Kocsor, A., (in Hungarian) Az MTBA magyar telefonbeszéd-adatbázis kézi feldolgozásának tapasztalatai, A Beszédkutatás 2003 kötetben (szerk. Gósy Mária), pp. 134-146, 2003.

[23] Tóth, L., Kocsor, A., Kovács, K., Felföldi, L.: (in Hungarian) Nyelvészeti tudásforrások integrálási lehetőségei diszkriminativ szegmens-alapú beszédfelismerő rendszerekbe, Proc. MSZNY 2003, pp. 169-174.

[24] Kocsor, A., Tóth, L.: Application of Kernel-Based Feature Space Transformations and Learning Methods to Phoneme Classification, Applied Intelligence, Vol. 21., No. 2., pp. 129-142, 2004.

[25] Kocsor, A., Tóth, L., Kernel-Based Feature Extraction with a Speech Technology Application, IEEE Trans. Signal Processing, Aug. 2004.

[26] Tóth, L., Gosztolya, G., Replicator Neural Networks for Outlier Modeling in Segmental Speech Recognition, accepted for ISNN 2004

THINNING ALGORITHMS FOR 3D BINARY PICTURES

Kálmán Palágyi and Attila Kuba

Thinning is an iterative object reduction technique that provides an approximation to the skeleton in a topology preserving way. It means that the result of the thinning process is topologically equivalent to the original object. Our attention is focused on developing 3D thinning algorithm that are to be used in medical applications. This report describes the skeleton, the major skeletonization techniques in the case of discrete objects, the thinning methodologies, and our results.

I. SKELETON AS A SHAPE FEATURE

Shape is a fundamental concept in computer vision. It can be regarded as the basis for high–level image processing stages concentrating on scene analysis and interpretation. There are basically two different approaches for describing the shape of an object:

- using the boundary that surrounds it and
- using the occupied region.

Boundary–based techniques are widely used but there are some deficiencies which limit their usefulness in practical applications especially in 3D. Therefore, the importance of the region–based shape features show upward tendency. The local object symmetries represented by the skeleton certainly cannot replace boundary–based shape descriptors, but complement and support them.

The skeleton is a region–based shape feature that has been proposed by Blum as the result of the Medial Axis Transform. A very illustrative definition of the skeleton is given using the prairie–fire analogy: the object boundary is set on fire and the skeleton is formed by the loci where the fire fronts meet and quench each others. The formal definition of the skeleton has been stated by Calabi: the skeleton of an object is the locus of the centers of all the maximal inscribed hyperspheres. The continuous skeleton of a solid 3D box is illustrated in Fig. 1.

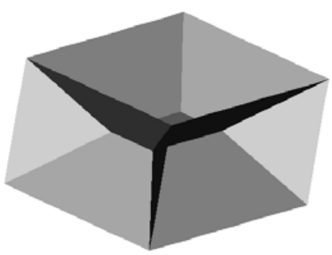

Figure 1. Example of 3D skeleton. The original object was a solid box.

II. SKELETONIZATION TECHNIQUES IN DISCRETE SPACES

During the last two decades skeletonization in the digital image raster has been an important research field. There are two major requirements to be complied with. The first one is geometrical. It means that the "skeleton" must be in the "middle" of the object and invariant under geometrical transformations. The second one is topological requiring that the "skeleton" must be topologically equivalent to the original object.

There are three major discrete skeletonization methods:

- based on distance transformation,
- thinning, and
- based on Voronoi–diagram.

We prefer thinning, since it:

- preserves topology,
- makes easy implementation possible (as a sequence of local Boolean operations),
- takes the least computational costs, and
- can be executed in parallel.

III. THINNING METHODOLOGIES

A 3D binary picture is a mapping that assigns value of 0 or 1 to each point with integer coordinates in the 3D digital space denoted by $\mathbb{Z}^3$. Points having the value of 1 are called black points, while 0's are called white ones. Black points form objects of the picture. White points form the background and the cavities of the picture. Both the input and the output of a picture operation are pictures. An operation is reduction if it can delete some black points (i.e., changes them to white) but white points remain the same. There is a fairly general agreement that a reduction operation is *not* topology preserving if any object in the input picture is split (into two or more ones) or completely deleted, if any cavity in the input picture is merged with the background or another hole, or if a cavity is created where there was none in the input picture. There is an additional concept called hole in 3D pictures. A hole (that doughnuts have) is formed by 0's, but it is not a cavity. Topology preservation implies that eliminating or creating any hole is not allowed.

Thinning must be a topology–preserving reduction. Existing 3D thinning algorithms can be classified from several points of view. One of them is the classification on the produced skeletons: some

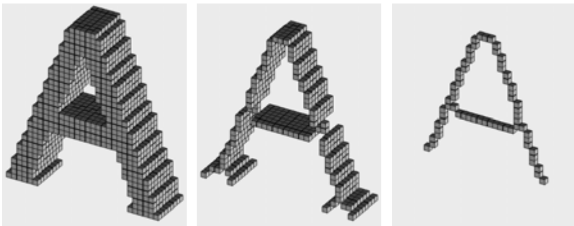

Figure 2. A 3D synthetic picture containing a character “A” (left); its medial surface (centre); its medial lines (right).

of the developed algorithms result in medial surfaces and others can produce medial lines (see Fig. 2).

Three types of parallel thinning methodologies have been proposed.

The first type examine the $3 \times 3 \times 3$ neighbourhood of each border point (i.e., black point that has at least one white point in its neighbourhood). The iteration steps are divided into a number of subiterations. Only border points having the prescribed $3 \times 3 \times 3$ neighbourhood can be deleted during a subiteration. It means that each prescribed neighbourhood gives a deletion condition. Prescribed neighbourhoods can be usually associated to a direction (e.g., up, down) depending on the position of the border points to be deleted. These algorithms use directional or border sequential strategy. Each subiteration is executed in parallel (i.e., all black points satisfying the deletion condition of the actual subiteration are simultaneously deleted). Most of existing parallel thinning algorithms are border sequential. Generally 6 subiterations are used.

A directional algorithm consisting of k subiterations follows the following process:

`repeat`
  `for` $i = 1$ `to` k `do`
    simultaneous deletion of the black points
    that satisfy the condition assigned to the
    i–th direction
`until` *no points are deleted*

We have developed three different border–sequential 3D thinning algorithms. They use 3, 6, 8, and 12 subiterations, respectively [1, 2, 3, 4].

The second type of algorithms does not need subiterations but — in order to preserve topology — it investigates larger ($5 \times 5 \times 5$) neighbourhood. This approach can be sketched by the following program:

`repeat`
  simultaneous deletion of the black points
  that satisfy the global condition
`until` *no points are deleted*

The third approach is the subfield sequential method. The set of points $\mathbb{Z}^3$ is divided into more disjoint subsets which are alternatively activated. At a given iteration step, only black points of the active subfield are designated to be deleted.

A subfield based algorithm consisting of k subfields can be described as follows:

`repeat`
  `for` $i = 1$ `to` k `do`
    simultaneous deletion of the black points
    in the i–th subfield that satisfy the global
    condition assigned to each subfield
`until` *no points are deleted*

We have proposed a new class of thinning methods called hybrid algorithms. It applies both directional and subfield strategies [5].

IV. APPLICATIONS

Thinning provides “skeleton–like” shape feature that are extracted from binary image data. It is a common preprocessing operation in raster–to–vector conversion or in pattern recognition. Its goal is to reduce the volume of elongated objects. Some important applications have been appeared in medical image processing (see Fig. 3), too [?].

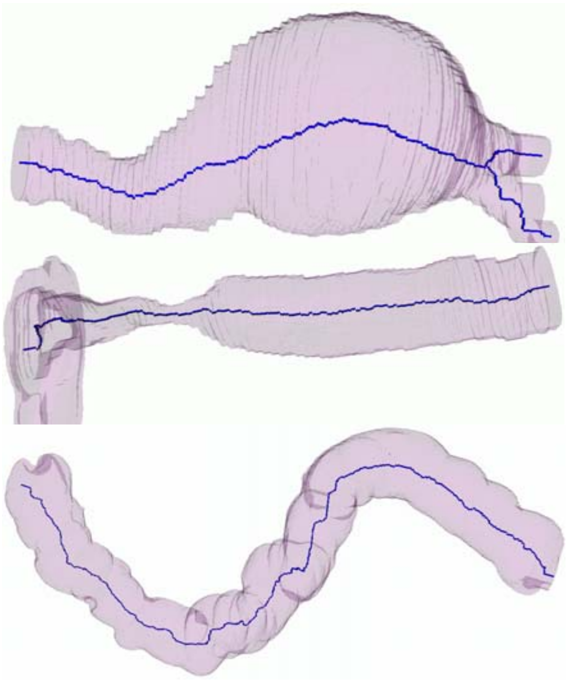

Figure 3. Central paths for a segmented part of an infrarenal aorta (top), a segmented upper respiratory tract (middle), and a segmented cadaveric phantom (bottom) produced by our curve-thinning algorithm. In order to get longer central paths, the endpoints were automatically identifie d before the thinning and those points were regarded as ”anchors“ during the thinning process (i.e., their deletion were prohibited). In that way, the topologically correct thinning algorithm was urged to connect them.

REFERENCES

[1] K. Palágyi, A 3-subiteration 3D thinning algorithm for extracting medial surfaces, Pattern Recognition Letters, to appear

[2] K. Palágyi, A. Kuba, A 3D 6–subiteration thinning algorithm for extracting medial

lines, Pattern Recognition Letters 19, 1998, 613-627.

[3] K. Palágyi, A. Kuba, A parallel 8–subiteration 3D thinning algorithm, in Proc. 8th Int. Conf. on Discrete Geometry for Computer Imagery, DGCI'99, Marne-la-Vallé, France, Lecture Notes in Computer Science 1568, Springer, 1999, 325-336.

[4] K. Palágyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm, Graphical Models and Image Processing 61, 1999, 199-221.

[5] K. Palágyi, A. Kuba, A hybrid thinning algorithm for 3D medical images, J. Computing and Information Technology 6, 1998, 149-164.

[6] K. Palágyi, E. Sorantin, Cs. Halmai, A. Kuba:, 3D thinning and its applications to medical image processing, TASK Quarterly 3, 1999, 397-408.

[7] K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdőhelyi, K. Hausegger, A sequential 3D thinning algorithm and its medical applications, in Proc. 17th Int. Conf. Information Processing in Medical Imaging, IPMI 2001, Davis, USA, Lecture Notes in Computer Science 2082, Springer, 2001, 409-415.

[8] E. Sorantin, Cs. Halmai, B. Erdőhelyi, L. Martonossy, K. Palágyi, B. Geiger, New advances for imaging of laryngotracheal stenosis by post processing of spiral-CT data, in W. Hruby (ed.) Digital Revolution in Radiology, Springer, 2000, pp. 275-285.

[9] E. Sorantin, E. Balogh, A. Vilanova i Bartroli, K. Palágyi, L.G. Nyúl, Virtual dissection of the colon based on helical CT data - Can it be done? in Proc. 2nd Int. Symposium on Image and Signal Processing and Analysis, ISPA 2001, Pula, Croatia, 2001, 224-229.

[10] E. Sorantin, G. Werkgartner, E. Balogh, A. Vilanova i Bartrolí, K. Palágyi, L.G. Nyúl, L. Ruskó, Virtual dissection and automated polyp detection of the colon based on spiral CT data - Techniques and preliminary experience on a cadaveric phantom, European Surgery - Acta Chirurgica Austriaca, 34, 2002, 143-151.

[11] E. Sorantin, Cs. Halmai, B. Erdőhelyi, K. Palágyi, L.G. Nyúl, L. K. Ollé, B. Geiger, F. Lindbichler, G. Friedrich, K. Kiesler, Spiral-CT-based assessment of tracheal stenoses using 3-D-skeletonization, IEEE Transactions on Medical Imaging, 21, 2002, 263-273.

[12] E. Balogh, E. Sorantin, L.G. Nyúl, K. Palágyi, A. Kuba, G. Werkgartner, E. Spuller, Virtual dissection of the colon: technique and first experiments with artificial and cadaveric phantoms, in Medical Imaging 2002: Image Processing, in Proceedings of SPIE Vol. 4681 , 2002, 713-721.

[13] J. Tschirren, K. Palágyi, J.M. Reinhardt, E.A. Hoffman, M. Sonka, Segmentation, skeletonization, and branchpoint matching - A fully automated quantitative evaluation of human intrathoratic airway trees, in Proc. 5th Int. Conf. Medical Image Computing and Computer-Assisted Intervention, MICCAI 2002, Part II, Lecture Notes in Computer Science 2489, Springer, 2002, 12-19.

[14] E. Sorantin, E. Balogh, A. Vilanova i Bartrolí, K. Palágyi, L.G. Nyúl, S. Loncaric, M. Subasic, D. Kovacevic, Virtual Dissection of the Colon, in D. Caramella, C. Bartolozzi (eds.) 3D Image Processing - Techniques and Clinical Applications, Springer, 2002, 197-209.

[15] K. Palágyi, J. Tschirren, M. Sonka, Quantitative analysis of three-dimensional tubular tree structures, in Medical Imaging 2003: Image Processing, in Proceedings of SPIE Vol. 5032, 2003, 277-287.

[16] K. Palágyi, J. Tschirren, M. Sonka, Quantitative analysis of intrathoracic airway trees: methods and validation, in Proc. 18th Int. Conf. Information Processing in Medical Imaging, IPMI 2003, Lecture Notes in Computer Science 2732, Springer, 2003, 222-233.

[17] E.A. Hoffman, J.M. Reinhardt, M. Sonka, B.A. Simon, J. Guo, O. Saba, D. Chon, S. Samrah, H. Shikata, J. Tschirren, K. Palágyi, K.C. Beck, G. McLennan, Characterization of the Interstitial Lung Diseases via Density-Based and Texture-Based Analysis of Computed Tomography Images of Lung Structure and Function, Academic Radiology, 10, 2003, 1104-1118.

TREE AUTOMATA AND TERM REWRITE SYSTEMS

Sándor Vágvölgyi

I. TERM REWRITE SYSTEMS EFFECTIVELY PRESERVING RECOGNIZABILITY

Tree automata and recognizable tree languages proved to be an efficient tool in the theory of term rewrite systems. Let Σ be a ranked alphabet, let R be a term rewrite system over Σ, and let L be a tree language over Σ. Then $R^*_\Sigma(L)$ is the set of descendants of trees in L. When Σ is apparent from the context, we simply write $R^*(L)$ rather than $R^*_\Sigma(L)$. A term rewrite system R over Σ preserves Σ-recognizability, if for each recognizable tree language L over Σ, $R^*(L)$ is recognizable. The signature $sign(R)$ of a term rewrite system R is the ranked alphabet consisting of all symbols appearing in the rules of R. Gilleron showed that for a term rewrite system R it is not decidable if R preserves $sign(R)$-recognizability. Otto showed that it is not decidable for a term rewrite system R whether R preserves recognizability. A term rewrite system R over Δ preserves recognizability, if for each ranked alphabet Σ with $sign(R) \subseteq \Sigma$, R preserves Σ-recognizability. We [7] showed that there is a ranked alphabet Σ and there is a linear term rewrite system R over Σ such that R preserves Σ-recognizability but does not preserve recognizability.

Let R be a term rewrite system over a ranked alphabet Σ. We say that R effectively preserves Σ-recognizability if for a given tree automaton $\mathcal{B}$ over Σ, we can effectively construct a tree automaton $\mathcal{C}$ over Σ such that $L(\mathcal{C}) = R^*_\Sigma(L(\mathcal{B}))$. Let R be a term rewrite system over a ranked alphabet Δ. We say that R effectively preserves recognizability if for any given ranked alphabet Σ with $sign(R) \subseteq \Sigma$ and any given tree automaton $\mathcal{B}$ over Σ, we can effectively construct a tree automaton $\mathcal{C}$ over Σ such that $L(\mathcal{C}) = R^*_\Sigma(L(\mathcal{B}))$.

In spite of the undecidability results of Gilleron and Otto, we know several term rewrite systems which preserve recognizability. Brainerd showed that ground term rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. Gallier and Book [4] introduced the notion of a monadic term rewrite system, and Salomaa showed that linear monadic term rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. A term rewrite system is monadic if each left-hand side is of depth at least 1 and each right-hand side is of depth at most 1. Coquidé et al. defined the concept of a semi-monadic term rewrite system generalizing the notion of a monadic rewrite system and the notion of a ground term rewrite system. A term rewrite system R over Σ is semi-monadic if, for every rule $l \to r$ in R, $depth(l) \geq 1$ and either $depth(r) = 0$ or $r = f(y_1, \ldots, y_k)$, where $f \in \Sigma_k$, $k \geq 1$, and for each $i \in \{1, \ldots, k\}$, either y_i is a variable (i.e., $y_i \in X$) or y_i is a ground term (i.e., $y_i \in T_\Sigma$). It is immediate that each monadic term rewrite system is semi-monadic as well. Coquidé et al. showed that linear semi-monadic term rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. We [7] generalized even further the concept of a semi-monadic term rewrite system introducing the concept of a generalized semi-monadic term rewrite system (gsm term rewrite system for short). We [7] showed that a linear gsm (lgsm) term rewrite system R over Δ effectively preserves recognizability.

Let R be a term rewrite system over $sign(R)$, and let $\Sigma = \{f, \sharp\} \cup sign(R)$, where $f \in \Sigma_2 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. We [7] showed that R effectively preserves Σ-recognizability if and only if R effectively preserves recognizability. We [8] improved this result for left-linear term rewrite systems. We showed the following. Let R be a left-linear term rewrite system over $sign(R)$, and let $\Sigma = \{g, \sharp\} \cup sign(R)$, where $g \in \Sigma_1 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. Then R effectively preserves Σ-recognizability if and only if R effectively preserves recognizability. This result makes it easier to show that a given left-linear term rewrite system effectively preserves recognizability. Furthermore, R preserves Σ-recognizability if and only if R preserves recognizability.

We [13] introduced and studied the half-monadic term rewrite system, which is an extension of the right-ground term rewrite system and is a slight extension of the semi-monadic term rewrite system. Takai, Kaji, and Seki introduced the concept of the finite path overlapping term rewrite system, and showed that right-linear finite path overlapping term rewrite systems effectively preserve recognizability. A half-monadic term rewrite system is a finite path overlapping term rewrite system as well. Hence right-linear half-monadic term rewrite systems effectively preserve recognizability. We [13] showed that termination and convergence are decidable properties for right-linear half-monadic term rewrite systems.

Let R be a right-linear half-monadic term rewrite system over a ranked alphabet Σ such that for every rule $l \to r$ in R, l is not a variable. We introduced two weight functions. Let $p \in T_\Sigma$ be a minimal height ground term starting an infinite reduction sequence. We introduced the weight function $\| \, \|_{p,R} : R^*(\{p\}) \to N$. For any $t \in R^*(\{p\})$, $\|t\|_{p,R}$ is the length of the longest reduction sequence starting from t, where we do not rewrite at the root and we do not count the rewrite steps of those immediate subtrees of the root which are subtrees of the descendants of the immediate ground subterms of the right-hand sides of the rules in R. We showed that $\to_R^*$ does not increase $\| \, \|_{p,R}$. We gave a condition which implies that $\to_R^*$ decreases $\| \, \|_{p,R}$. For an arbitrary $p \in T_\Sigma$, we introduced the weight function $\langle \, \rangle_{p,R} : R^*(\{p\}) \to N$. For any $t \in R^*(\{p\})$, $\langle t \rangle_{p,R}$ is the number of occurrences of those subtrees which are not subtrees of the descendants of the immediate ground subterms of the right-hand sides of the rules in R. We showed that $\to_R^*$ does not increase $\langle \, \rangle_{p,R}$. We gave a condition which implies that $\to_R^*$ decreases $\langle \, \rangle_{p,R}$.

Using the effective preserving recognizability property of R, we show that for a ground term $t \in T_\Sigma$ it is decidable if $t \in T_\Sigma$ starts an infinite reduction sequence. Using weight functions $\| \, \|_{p,R}$ and $\langle \, \rangle_{p,R}$, we constructed a finite set $W \subseteq T_\Sigma$ of terms such that R is terminating if and only if no term in W starts an infinite reduction sequence. We constructed W in two steps. First we put in W the subtrees of the descendants of the ground subterms of the right-hand sides of the rules in R. Then we put in W all ground terms obtained from the left-hand sides of the rules in R by replacing the variables with nullary symbol \$ and subtrees of the descendants of the ground subterms of the right-hand sides of the rules in R. In this way we decide if R is terminating. Hence we get that termination is a decidable property for right-linear half-monadic term rewrite systems. As R preserves recognizability we can decide if R is locally confluent. Hence convergence is a decidable property for right-linear half-monadic term rewrite systems.

II. GROUND TERM REWRITE SYSTEMS

In paper [9], we showed that a reduced ground term rewrite system over some ranked alphabet Σ has a good property: if we omit any rules, the omitted rules and the remaining rules generate congruence relations such that their intersection is the identity relation on T_Σ. In general, given two ground term rewrite systems A and B over some ranked alphabet Σ with $\leftrightarrow_A^* \subseteq \leftrightarrow_B^*$, one may want to eliminate all nontrivial pairs of $\leftrightarrow_A^*$ from $\leftrightarrow_B^*$. In other words, one may want to construct a ground term rewrite system (gtrs for short) C over Σ such that $\leftrightarrow_{A \cup C}^* = \leftrightarrow_B^*$ and that $\leftrightarrow_A^* \cap \leftrightarrow_C^*$ is the identity relation on T_Σ. Let A and B be gtrs's over some ranked alphabet Σ with $\leftrightarrow_A^* \subseteq \leftrightarrow_B^*$. We say that a gtrs C over Σ is a congruential complement of A for B, if $\leftrightarrow_{A \cup C}^* = \leftrightarrow_B^*$ and $\leftrightarrow_A^* \cap \leftrightarrow_C^*$ is the identity relation over T_Σ. We note that A may have more than one congruential complement for B, which are pairwise nonequivalent. Our [9] main result was the following. Given gtrs's A, B, C over some ranked alphabet Σ with $\leftrightarrow_A^* \subseteq \leftrightarrow_B^*$, one can effectively decide if C is a congruential complement of A for B.

Dauchet et al. have introduced the notion of the ground tree transducer as a pair (A, B) of tree automata. The importance of ground tree transducers is in that they can simulate ground term rewriting: it was shown that for each ground term rewriting system R over a ranked alphabet Σ, one can effectively construct a ground tree transducer (A, B) over Σ such that $\to_R^*$ is equal to the tree transformation $\tau_{(A,B)}$ induced by (A, B). M. Dauchet, P. Heuillard, P. Lescanne and S. Tison have used this result to show that the confluence property of ground term rewrite systems is decidable. From now on we write $\pi(A)$ for $\tau_{(A,A)}$.

In paper [6] we considered restricted versions of ground tree transducers. Our motivation was that studying restricted versions of a class of machines frequently gives a deeper insight into the working of the unrestricted class. As usual for a class of tree transducers, we considered the total and the deterministic subclasses. We call a ground tree transducer (A, B) deterministic (total) if the tree automata A and B are deterministic (total). We also considered symmetric ground tree transducers, which are of the form (A, A). Moreover, we considered the eight classes of ground tree transducers obtained by combining these three properties in all possible ways.

As the first result, we compared the expressive power of the eight classes by presenting the full inclusion diagram of the tree transformation classes induced by them. Then we [6] showed that the following four classes of term relations are the same: (i) the tree transformations induced by symmetric deterministic ground tree transducers, (ii) the congruence relations on term algebras induced by reduced ground term rewriting systems, (iii) the congruence relations on term algebras induced by convergent ground term rewriting systems, and (iv) the finitely generated congruence relations on term algebras.

In [3], Fülöp and Vágvölgyi showed the following. For every ground term equation system E over a ranked alphabet Σ, which is just a finite binary relation on T_Σ, one can effectively construct a deterministic tree automaton A over Σ such that $\leftrightarrow_E^*$, i.e., the congruence relation induced by E on the Σ-term algebra, is equal to the tree transformation $\pi(A)$ induced by the symmetric deterministic ground tree transducer (A, A).

In the proof of the inclusion (i) $\subseteq$ (ii) we constructed, for a given deterministic tree automaton A over a ranked alphabet Σ, a reduced ground term rewriting system R over Σ such that the congruence relation $\leftrightarrow_R^*$ generated by R is equal to the tree transformation $\pi(A)$.

Thus, as a by-product of our results, we obtained a new ground completion algorithm. Given ground term equation system E, we constructed a reduced ground term rewriting system equivalent to E in two steps. In the first step, presented in [3], we computed a symmetric deterministic ground tree transducer (A, A) such that $\pi(A) = \leftrightarrow_E^*$. Then we constructed as in the proof of (i) $\subseteq$ (ii) a reduced ground term rewriting system R such that $\pi(A) = \leftrightarrow_R^*$. Hence $\leftrightarrow_E^* = \leftrightarrow_R^*$. This ground completion parallels to Snyder's fast algorithm.

It can easily be proved that a binary relation τ over T_Σ can be induced by a total symmetric deterministic ground tree transducer if and only if τ is a congruence of finite index on the Σ-term algebra. In addition, we [6] showed that any finitely generated congruence relation τ over the Σ-term algebra is of finite index if $trunk(\tau) = T_\Sigma$. The concept of the trunk was introduced in [5] and proved to be useful in studying congruence relations over terms. Furthermore, we proved that for a term algebra with at least one non-nullary function symbol the congruence relations of finite index are exactly the finitely generated congruence relations with trunks equal to the set of all terms.

Fülöp and Vágvölgyi [6] constructed, for a given deterministic tree automaton A over Σ, a reduced ground term rewrite system R over Σ such that $\leftrightarrow_R^*$ is equal to the ground tree transformation $\pi(A)$. They also showed that the class of ground tree transformations induced by tree automata properly contains the class of ground tree transformations induced by deterministic tree automata.

A tree automaton A over Σ induces another binary relation, $\theta(A) = \leftrightarrow_A^* \cap T_\Sigma \times T_\Sigma$ over T_Σ. It is the restriction of the congruence $\leftrightarrow_A^*$ on the $\Sigma \cup A$-term algebra to terms over Σ. Hence $\theta(A)$ is a congruence on the Σ-term algebra. Fülöp and Vágvölgyi [2] showed the following. For every ground term rewrite system R over Σ, one can effectively construct a deterministic tree automaton A over Σ such that $\leftrightarrow_R^* = \theta(A) = \pi(A)$. We [12] showed that for any deterministic tree automaton A over Σ, $\theta(A) = \pi(A)$.

For a tree automaton A over Σ, we [12] defined a congruence relation on A, called the determiner of A. We showed that we can effectively compute the determiner ρ, and construct the quotient tree automaton A/ρ. We called A/ρ the principal quotient of A, and showed that A/ρ is a deterministic tree automaton. We [12] showed the following results.

- For any tree automata A and B over Σ and homomorphism $\phi : A \to B$, $\theta(A) \subseteq \theta(B)$ and $\pi(A) \subseteq \pi(B)$.
- For any tree automaton A, the following two statements are equivalent.

  (i) $\pi(A) = \pi(A/\rho)$.

  (ii) There is a deterministic tree automaton B such that $\pi(A) = \pi(B)$.
- For any connected tree automaton A, Conditions (i) and (ii) are equivalent to Condition (iii):

  (iii) $\theta(A) = \pi(A)$.
- For any tree automaton A over Σ, Conditions (i), (ii), and (iii) are decidable.

We [10] showed that it is decidable for any given ground term rewrite systems R and S if there is a ground term rewrite system U such that $\leftrightarrow_U^* = \leftrightarrow_R^* \cap \leftrightarrow_S^*$. If the answer is yes, then we can effectively construct such a ground term rewrite system U. In other words, for any given finitely generated congruences ρ and τ over the term algebra, it is decidable if $\rho \cap \tau$ is a finitely generated congruence. If the answer is yes, then we can effectively construct a ground term rewrite system U such that $\leftrightarrow_U^* = \rho \cap \tau$.

In [11] we studied rewriting of ground terms. For an arbitrary term rewrite system R over a ranked alphabet Σ, we restricted the rewriting relation $\to_R$ to ground terms. We introduced the relation $\to_{R,g} = \to_R \cap (T_\Sigma \times T_\Sigma)$. We showed that $\leftrightarrow_{R,g}^* = \leftrightarrow_R^* \cap (T_\Sigma \times T_\Sigma)$. We showed that for a given term rewrite system R and a given ground term rewrite system S over a ranked alphabet Σ it is decidable if $\leftrightarrow_{R,g}^* \subseteq \leftrightarrow_S^*$. We showed that for a given left-linear right-ground term rewrite system R over a ranked alphabet Σ it is decidable if there is a ground term rewrite system S over Σ such that $\to_{R,g}^* = \to_S^*$. If the answer is yes, then one can effectively construct a gtrs S over Σ such that $\to_{R,g}^* = \to_S^*$.

In paper [1] we studied that version of the ground tree transducer game where the same tree automaton appears as the first and second component of the associated ground tree transducer. We gave conditions which imply that Beta has a winning strategy. Furthermore, we showed the following decidability result. Given a ground tree transducer game where the underlying tree automaton A cannot evaluate some tree into a state or A is deterministic, we can decide which player has a winning strategy. Moreover, whatever player has a winning strategy, we can effectively construct a partial recursive winning strategy for him.

ACKNOWLEDGEMENTS

The mentioned research tasks were supported by the following grants: FKFP 0095/1997 of the Hungarian Cultural and Educational Ministry, OTKA T022994, OTKA T030084 of the Research Foundation of Hungary, the Bolyai János Research Scholarship of the Hungarian Academy of Sciences, and the Széchenyi István Scholarship of the Hungarian Ministry of Education.

REFERENCES

[1] J. Apró and S. Vágvölgyi, The ground tree transducer game with identical tree automata, *Fundamenta Informaticae*, **45** (2001) 157-172.

[2] Z. Fülöp and S. Vágvölgyi, Congruential tree languages are the same as recognizable tree languages - A proof for a theorem of D. Kozen, *Bulletin of the EATCS*, **39** (1989) 175-185.

[3] Z. Fülöp and S. Vágvölgyi, Ground term rewriting rules for the word problem of ground term equations, *Bulletin of the EATCS*, **45** (1991) 186-201.

[4] J.H. Gallier and R.V. Book, Reductions in tree replacement systems *Theoretical Computer Science* **37** (1985) 123-150.

[5] Z. Fülöp and S. Vágvölgyi, Minimal Equational Representations of Recognizable Tree Languages, *Acta Informatica*, **34** (1997) 59-84.

[6] Z. Fülöp and S. Vágvölgyi, Restricted Ground Tree Transducers, *Theoretical Computer Science*, **250** (2001) 219-233.

[7] P. Gyenizse and S. Vágvölgyi, Linear Generalized Semi-monadic Rewrite Systems Effectively Preserve Recognizability, *Theoretical Computer Science*, **194** (1998) 87-122.

[8] P. Gyenizse, S. Vágvölgyi, A property of left-linear rewrite systems preserving recognizability, *Theoretical Computer Science*, **242** (2000) 477-498.

[9] S. Vágvölgyi, Congruential Complements of Ground Term Rewrite Systems, *Theoretical Computer Science*, **238** (2000) 247-274.

[10] S. Vágvölgyi, Intersection of Finitely Generated Congruences over Term Algebra *Theoretical Computer Science*, **300** (2003) 209-234.

[11] S. Vágvölgyi, Term Rewriting Restricted to Ground Terms *Theoretical Computer Science*, **302** (2003) 135-165.

[12] S. Vágvölgyi, On Ground Tree Transformations and Congruences Induced by Tree automata *Theoretical Computer Science*, **304** (2003) 449-459.

[13] S. Vágvölgyi, Right-Linear Half-Monadic Term Rewrite Systems, *Theoretical Computer Science*, **309** (2003) 195-211.