Fourth Biennial Report

April 1997 – April 1999

Technical Annex

Contents

Part I

Overview – The Institute

Summary

The institute made a major step forward in the last two years and expanded from two to four research units. Two new directors joined the institute:

- Thomas A. Henzinger started on January 1, 1999. He directs the research unit on Reactive and Hybrid Systems.
- Hans-Peter Seidel starts on June 1, 1999. He directs the research unit on Computer Graphics.

The existing groups for Algorithms and Complexity (Kurt Mehlhorn) and Programming Logics (Harald Ganzinger) continued with their successful work.

Research Program

The institute is devoted to basic research in computer science, and in particular to the study of complex computer systems. Complexity in computer systems arises for various reasons:

A problem can be complex due to huge masses of data that have to be handled, sometimes in real time. For this sort of problem efficient algorithms and data structures as well as parallel processing are of great importance. Kurt Mehlhorn's group addresses this facet of complexity.

Complexity can mean logical complexity as we find it in large software systems, with many layers of abstraction, where applications from different problem domains interact with each other in often unpredictable ways. Here we need to apply methods based on mathematical logic in order to structure, reason about, and develop more systematically, such large systems. Harald Ganzinger's group addresses this facet of complexity.

Today's computer systems frequently consist of many interacting processes, which are often embedded into a natural environment that is governed by physical laws. Here complexity arises due to concurrency, real-time behavior, and heterogeneity (mixed hardware-software, mixed synchrony-asynchrony, mixed discrete-continuous behavior). Methods for understanding and controlling these sources of complexity rely on a combination of algorithmic, logical, automata-theoretic, and game-theoretic techniques. Thomas Henzinger's group addresses this facet of complexity.

Computer systems are more and more used to realize and simulate parts of the real or an imaginary world. Such simulations require to model, to render, and to animate complex objects. The goal of computer graphics is to turn abstract information into visual images and to allow the user to interact with complex objects and data in a natural and intuitive way. Hans-Peter Seidel's group addresses this facet of complexity.

Structure

The institute is planned to consist eventually of five research groups, to fit this research program. Currently there are four research units. Potential areas that have been identified for the remaining unit include distributed and fault-tolerant computing, computer networks, robotics, computer vision, and multi-media systems.

The establishment of two new research units has substantially decreased the "average distance" between the research units. One of the challenges of the next years will be to turn the decreased average distance into fruitful cooperations.

At present 18 research associates, 25 doctorate students and 14 postdocs are affiliated with the institute. The scientific staff is complemented by an administrative unit with 15 members (including secretaries), by a computing support unit (5 members of staff) and by our library (2

members of staff). The computing support unit currently operates a network of approximately 200 workstations.

Grants

The institute is involved in a number of projects related to research grants awarded by the European Union, by the German Science Foundation (DFG), by the German Ministry for Education and Research (BMBF), and by industry; for the descriptions of these grants see sections III.14 and IV.14. Funding of these projects in 1997 was about 788 000 DM (649 000 DM in 1998).

Results

In the parts to follow we describe in detail, for the Algorithms and Complexity group (Kurt Mehlhorn) and the Programming Logics Group (Harald Ganzinger), the research programs and results obtained in the period May 1997 through April 1999. We also briefly discuss the work of the two new groups. We have continued to be very successful in our research, as documented by our many scientific publications, including about 330 articles in journals, books or proceedings of major international conferences.

Many of the institute's results are, in addition, available to the public through computer programs such as the LEDA library of efficient algorithms, and the SPASS theorem prover for first-order logic.

Teaching activities

The institute makes an effort to offer a variety of courses to computer science students of the Universität des Saarlandes. Courses taught by members of the two "old" groups during the period of this report are listed in Sections III.11 and IV.12; Thomas Henzinger teaches a course on Computer-Aided Verification in this semester. In the period of this report 8 doctoral dissertations and 5 habilitations have been successfully completed.

Professional activities

Members of the institute have been involved in the organization of 21 workshops and conferences. In 20 cases we have been invited to join the program committee of major international conferences, not counting program committee memberships for national and international workshops. Finally, we serve on the editorial board of 12 scientific journals.

Part II

Overview – The Research Units

1 The Algorithms and Complexity Group

The "Algorithms and Complexity" group

- investigates the inherent complexity of computational problems,
- designs and analyses efficient algorithms for fundamental combinatorial and geometric problems (both for sequential and parallel machines),
- develops software libraries to bridge the gap between algorithms research and the use of algorithms, and
- solves algorithmic problems in specific domains, such as computational biology and graph visualization.

We work theoretically and we develop software. About two-thirds of our resources go into theoretical work and about one-third goes into software development. KM[1] believes that this is a healthy balance. Our theoretical research has led to numerous publications in reputed conferences (STOC, FOCS, ESA, SODA, CompGeo, RECOMB, Graph Drawing, IPCO, COCOON, ICALP) and journals (see Section 15) and the practical work has resulted in widely used commercial-quality software. In this way the group has a standing in the theory community, but also has impact far beyond theoretical computer science. We are also very active in teaching.

The group consists of research associates, postdocs, and Ph.D. students. Research associates are typically on five year contracts (extendible by two years under certain circumstances), and postdocs stay for either one or two years (unless they continue as a research associate). We will see a major change in the senior personnel of the group in the next two years. Torben Hagerup left the group to become Professor of Computer Science in Frankfurt. Susanne Albers, Petra Mutzel, Jop Sibeyn, and Rudolf Fleischer completed their Habilitation procedures and the procedures of Hans-Peter Lenhof and Stefan Schirra are on-going. Three of them already got offers for professorships.

Most of the researchers in the group are on two-year postdoc contracts or three-year graduate student scholarships and hence there is considerable fluctuation. We run an intensive seminar and lecture program to spur interaction within the group. We run six special interest group seminar series which typically meet every other week, we have a "noon seminar" for topics of general interest, and we run a course on selected topics in algorithms. There are currently 18 Ph.D. students working in the group. We expect at least six of them to finish in 99. We have to admit, however, that only one graduate student (U. Finkler) finished his Ph.D. work in the last 18 months. We have an intensive visitor program; more than 57 guests visited our group for stays up to three months, cf. Section 9 for details.

In this report we discuss our work under the following headings:

- Data Structures and Graph Algorithms
- On-line and Approximation Algorithms
- Parallel and External Computing
- Computational Geometry
- Computational Molecular Biology and Chemistry

[1] In this summary KM stands for Kurt Mehlhorn.

- Graph Drawing
- Software Libraries

We have maintained our activities in computational geometry, parallel algorithms, data structures and graph algorithms, and the construction of software libraries. We have further increased our strength in combinatorial optimization (on-line and approximation algorithms and the use of the branch-and-cut method for the solution of NP-complete problems), graph drawing, and computational biology and chemistry. The activity in external memory algorithms is new and our activity in computational complexity is subsumed in the other activities. We now briefly survey our results in each of the areas and highlight some of the results. We concentrate on results obtained by Postdocs and Ph.D. students. Most of our research associates are competing for professorships in the moment and so KM feels it would be inappropriate to highlight their work in a publicly accessible document. We also list the researchers, postdocs, and Ph.D.-students that are currently (May 1st, 1999) working in the area and the fraction of their time that is devoted to the area. In case of outside funding this is also indicated[2]. We use different organizational schemes for the different areas: tightly coordinated groups in all areas that involve significant amounts of implementation and loosely connected sets of researchers in the other areas. In the former case we will also name the coordinator(s) of the area.

Data Structures and Graph Algorithms: This is the smallest of our research areas. We investigated classical data structuring problems like the dictionary problem, flow, cut and shortest path problems in graphs, tree decomposition, and matroid optimization problems.

Our Postdoc C.R. Subramanian and our Ph.D. student Volker Priebe worked on the average case complexity of shortest path problems. C.R. investigated the $G(n,p)$ model in which any edge in an n-vertex graph is present with probability p and each existing edge has a random weight in the range $\{0,\ldots,n\}$ and showed that the all-pairs shortest path problem can be solved in time $O(n^2 \log n)$ with high probability. The proof is based on an extension of Jensen's inequality. Volker obtained the same time bound for a completely different model. Each edge of the complete graph has a non-negative random weight (which is not revealed to the algorithm) and each node has a node potential fixed arbitrarily. The reduced weights of the edges with respect to the potential are revealed. In this model edges may have negative weights, but there are no negative cycles.

Research Associates: Kurt Mehlhorn (1/10), Rudolf Fleischer (3/10)
Postdocs: Zeev Nutov (3/10), Peter Sanders (1/10), Steven Seiden (1/20), Roberto Solis-Oba (3/10), C.R. Subramanian (1/1)
Ph.D. students: Ernst Althaus (1/3), Andreas Crauser (1/4), Volker Priebe (1/1), Mark Ziegelmann (3/10, Graduiertenkolleg), Joachim Ziegler (1/5)

On-line and Approximation Algorithms: Our activity in this area has largely increased in the last two years. We have worked on a wide range of scheduling problems, graph problems, and packing problems.

Our Postdocs Klaus Jansen and Roberto Solis-Oba made significant progress for the job shop scheduling problem. The goal is to schedule n jobs on m machines. Each job consists of μ tasks. The tasks of each job must be executed in order and for each task the duration and the machine is

[2] ALCOM-IT and GALIA are projects financed by the European Union, DFG stands for Deutsche Forschungsgemeinschaft (German Research Foundation), BMBF stands for Ministry of Education and Research, and Graduiertenkolleg stands for the special Ph.D. program financed by the DFG.

prescribed. Klaus and Roberto obtained a linear time approximation scheme for the case of fixed m and μ. They divide the set of jobs into large (only a constant number) and small jobs and find for each relative ordering of the large jobs a (fractional) schedule for the small jobs using linear programming. The fractional schedule is rounded and then massaged into a feasible solution.

Our Ph.D. student Jordan Gergov designed approximation algorithms for compile-time memory allocation. Given a set of objects of different sizes and different life-spans, the goal is to allocate the objects to memory such that the objects do not interfere and the total amount of memory is minimized. Jordan designed a 3-approximation algorithm for the problem.

Research Associates: Susanne Albers (1/1), Rudolf Fleischer (2/5)
Postdocs: Zeev Nutov (7/10), Lorant Porkolab (4/5), Steven Seiden (19/20), Roberto Solis-Oba (3/5)
Ph.D. students: Jordan Gergov (1/1), Piotr Krysta (1/1, Graduiertenkolleg)

Parallel and External Computing: Our effort in this area has been refocused. We took up external memory computing as a new subject (more generally, computing that takes the memory hierarchy of modern processors into account). The connection between parallel and external computing is twofold. There is a connection on the level of methods: a surprising number of techniques that proved useful in the parallel context are also useful in the external memory setting. The second connection is on the level of motivation: Problem instances requiring external memory computation are also natural candidates for parallel computation. We developed parallel algorithms for interconnection networks and PRAMs (for sorting, list ranking, and graph problems) and we developed external memory algorithms for data base search, priority queues, and basic geometric problems.

Our Postdoc Ka Wong Chong designed a time and work optimal EREW-PRAM algorithm for the minimum spanning tree problem, thus answering a long-standing open problem.

Our Postdoc Peter Sanders and our Ph.D. student Uli Meyer contributed to parallel algorithms for the shortest path problem, to external memory priority queues, and to external memory algorithms for randomized incremental constructions. Our Ph.D. student Andreas Crauser developed a LEDA extension for secondary memory computations.

Research Associates: Kurt Mehlhorn (3/10), Jop F. Sibeyn (1/1)
Postdocs: Peter Sanders (9/10), Roberto Solis-Oba (1/20)
Ph.D. students: Hannah Bast (1/1), Stefan Burkhardt (1/10, Graduiertenkolleg), Andreas Crauser (1/2), Ulrich Meyer (1/1, Graduiertenkolleg), Mark Ziegelmann (3/10, Graduiertenkolleg),

Computational Geometry: Our work ranges from the theoretical investigation of fundamental geometric problems to the theoretical basis for implementations of geometric algorithms.

Our Postdoc Lorant Porkolab showed that semidefinite programming in fixed dimensions can be done in linear time. This extends Megiddo's result for linear programming. He also showed that it is possible to decide whether a semi-algebraic set in fixed dimensions contains an integral point. This extends Lenstra's result for integer linear programming. Both results were obtained jointly with his former advisor L. Khachyan. Lorant's results could alternatively be listed under computational complexity.

Our Postdoc Edgar Ramos obtained improved algorithms for fundamental geometric problems such as range reporting and ray shooting and he also derived a new curve reconstruction algorithm (with long-term guest Tamal Dey).

Our Ph.D. students Stefan Funke and Michael Seel much improved the floating filter technology

required for the exact and efficient implementation of geometric algorithms (together with Christoph Burnikel).

Research Associates: Kurt Mehlhorn (3/10), Christoph Burnikel (1/3), Stefan Schirra (1/3, GALIA)
Postdocs: Susan Hert (1/2), Lorant Porkolab (1/5), Edgar Ramos (1/1), Roberto Solis-Oba (1/20)
Ph.D. students: Ernst Althaus (1/3), Stefan Funke (1/2, Graduiertenkolleg), Michael Seel (1/2, ALCOM-IT), Mark Ziegelmann (3/10, Graduiertenkolleg)

Computational Molecular Biology and Chemistry: This effort is headed by Hans-Peter Lenhof. The work in this area is applied in the sense that the problems to be considered are defined by chemists and biologists and are solved in close cooperation with them. The cooperation partners are the MPIs for Molecular Physiology and Enzymologie of the Protein Folding Process and chemists at the Universität des Saarlandes.

We worked on the sequence alignment problem, the protein docking problem, data base search algorithms, parallel algorithms for molecular dynamics of synthetic molecules, and the biochemical algorithms library BALL.

Our Ph.D. student Knut Reinert developed algorithms for sequence alignment based on the branch-and-cut paradigm. His algorithms are considerably better than algorithms based on dynamic programming.

Our Ph.D. student Peter Müller developed parallel algorithms for the molecular dynamics simulation of synthetic molecules that reach almost optimal speed-ups for a medium number (less than forty) of processors.

Our Ph.D. student Stefan Burkhardt is one of the main contributors to an improved algorithm for high similarity searches in DNA databases. The algorithm is more than an order of magnitude faster than algorithms in widespread use.

Research Associates: Hans-Peter Lenhof (1/1), Christine Rüb (1/1)
Ph.D. students: Stefan Burkhardt (9/10, Graduiertenkolleg), Oliver Kohlbacher (1/3, DFG), Peter Müller (1/1)

Graph Drawing: This effort is headed by Petra Mutzel. Most of the work of the graph drawing group is driven by the following thesis: Although most of the interesting problems in graph drawing are NP-complete, they can be solved to optimality because the problem instances arising in practice are fairly small. Optimal solutions look much better than sub-optimal solutions. The technical vehicle to produce optimal solutions is the branch-and-cut approach to integer programming.

Many planar graph drawing problems are NP-hard in the case that the combinatorial embedding is not part of the input and can be solved in polynomial time if the combinatorial embedding is fixed. Our PhD-student René Weiskircher has given an elegant characterization for the set of combinatorial embeddings which gives the possibility of optimization in a variable embedding setting.

Our Ph.D. student Gunnar Klau has developed an algorithm for two-dimensional compaction of drawings. Two-dimensional compaction was a dream of the VLSI-community many years ago. It is feasible for graph drawing because there is more structure which can be exploited.

The graph drawing group has developed AGD, a library for automatic graph drawing.

Research Associates: Rudolf Fleischer (1/10), Petra Mutzel (7/8)
Ph.D. students: Gunnar Klau (7/10, BMBF), René Weiskircher (1/1, Graduiertenkolleg), Thomas Ziegler (9/10, Siemens)

Software Libraries: We are involved in the construction of four software libraries: LEDA, CGAL, BALL, AGD.

We have continued our work on the LEDA platform of combinatorial and geometric computing. All work on LEDA is joint work with Stefan Näher's group in Halle and Kurt and Stefan have finally finished their book about the LEDA system. In the course of writing the book the efficiency of many algorithms has been improved and many program checkers have been developed. The functionality of the library was extended by extension packages for abstract Voronoi diagrams, Steiner trees, graph iterators and tools for tuning geometric computations.

The work on CGAL (Computational Geometry Algorithms Library) is directed by Stefan Schirra and is part of the EU-project GALIA. CGAL is based on the software paradigm of generic programming and offers unmatched flexibility and functionality for geometric computing.

BALL (biochemical algorithms library) is an object-oriented framework for rapid software prototyping in molecular modeling and related areas. It is the Ph.D. project of Oliver Kohlbacher. It has already greatly decreased our own software development time.

AGD (Automatic Graph Drawing) is the outgrowth of our work on graph drawing. It offers a large number of graph drawing algorithms. Its distinguishing feature is the ability to solve many of the NP-complete problems arising in graph drawing to optimality (at least for the moderate size instances of interest).

Licenses for LEDA, CGAL, and AGD are marketed by our spin-off company Algorithmic Solutions GmbH. Our software is used in more than 1500 academic institutions and has been licensed to more than 100 companies.

Research Associates: Kurt Mehlhorn (3/10), Christoph Burnikel (2/3), Rudolf Fleischer (1/5), Petra Mutzel (1/8), Stefan Schirra (2/3, GALIA)
Postdocs: Susan Hert (1/2)
Ph.D.-students: Ernst Althaus (1/3), Andreas Crauser (1/4), Stefan Funke (1/2, Graduiertenkolleg), Gunnar Klau (3/10, BMBF), Oliver Kohlbacher (2/3, DFG), Michael Seel (1/2, ALCOM-IT), Mark Ziegelmann (1/10), Joachim Ziegler (4/5), Thomas Ziegler (1/10, Siemens)

The group contributes to the master's program in Computer Science at the Universität des Saarlandes. We have offered more than 25 courses, seminars, and project classes in the past two years, cf. Section 11 for details.

The group is involved in several national and international research projects. Section 14 gives details. We find it particularly pleasing that several of our research associates have won own research grants from the DFG or the BMBF.

Our scientific work is discussed in more detail in Sections 2 through 8. To be easily distinguishable, references regarding work done by members of the group (during the period considered in this report) are marked with a •.

2 The Programming Logics Group

The research unit "Programming Logics" applies methods of mathematical logic to a variety of problems in computer science. Computation is deduction, a principle that is taken literally in the area of Logic Programming. Formal specifications of software and hardware are formulae in logical systems. Program development and verification is based on proving theorems about specifications and programs. Solving problems in specific application domains such as mathematical optimization or program analysis can be seen as deduction with respect to specific constraint structures such as the integers, reals or Herbrand terms.

Computation often means to simulate some model of the real world. While in logic programming and in program synthesis and verification one applies, to a large extent, the classical logics known from mathematics, simulation of the real world requires logics that allow to treat incomplete and changing knowledge and to reason about beliefs, wishes, knowledge, and the like, of their agents. In that contextnon-classical logics have to be designed, investigated and applied.

Our work is both theoretical and practical in nature. A large fraction of it is essentially concerned with searching for new and better methods for finding proofs with the support of a computer. As the practical worth of results in this area can often not be judged from the theory alone, we are engaged in various implementation projects in which we try to obtain experimental evidence of the practical potential of our results.

During the last two years, the research unit has changed in its structure. The key members of the group on higher-order logic and logical frameworks, Basin, Matthews, and Viganò have left MPI, so that research in this area is presently discontinued. In exchange, Andreas Podelski has started to build up a new group on logic methods for program analysis. The group focuses on the investigation of the fundamental logical structures that arise from modelling program analysis frameworks and on the construction and application of suitable deductive methods.

Automated Theorem Proving for Predicate Logic (Coordinators: Harald Ganzinger, Christoph Weidenbach)

The SPASS theorem prover, an implementation of superposition with automatically inferred types, has been further developed in a team lead by Christoph Weidenbach. SPASS has continued to hold its excellent rank among the best ATP systems world-wide. At the last three CADE competitions of ATP systems, SPASS won 4 first prices, more than any other of the competing systems. We have started to use SPASS for the analysis of security protocols.

On the theoretical side, a number of very interesting new results have been obtained. Many of them are related to the decision problem and complexity analysis for certain fragments of first-order logic. For instance, Ganzinger and de Nivelle have shown that the [loosely] guarded fragment with equality can be decided by a surprisingly simple instance of superposition, and that the decision procedure is theoretically optimal. Ganzinger, Hustadt, Meyer, and Schmidt have succeeded in constructing a decision procedure for certain (modal) logics with transitive (possibly non-symmetric) relations as an instance of ordered chaining. These results are significant as they allow us to apply our SPASS and SATURATE provers to the satisfiability problem for these logics. Bachmair and Ganzinger have also solved a problem that was open for about 10 years regarding the refutational completeness of a certain superposition calculus proposed by Zhang and Kapur.

We also have continued our investigation into the combination of algebraic and logic methods for theorem proving in commutative algebraic theories such as Abelian groups, rings, and lattices. Waldmann was able to demonstrate the usefulness of his inference systems for torsion-free Abelian groups by showing that they yield a decision procedure for the word problem in these theories.

The main problem in designing new proof calculi is the construction of suitable termination orderings upon which proofs of redundancy for unwanted inferences can be based. An extension of the associative path ordering to more general algebraic theories found by Stuber turned out to be very useful in this regard. In this area, Sofronie-Stokkermans has considerably extended our methodological repertoire by exploiting representation theorems, in particular, Priestley duality.

Logic Methods for Program Analysis (Coordinator: Andreas Podelski)

We are developing mainly two kinds of methods for the analysis of programs with symbolic and numeric data, respectively. Although the methods have different uses and require substantially different algorithms, the abstract view is the same: express a program property as a specific solution of a constraint over sets of values and compute the solution by inferring a logically equivalent constraint in solved form.

Charatonik and Podelski have shown how one can express a CTL state property of a while program over trees as a solution of a set constraint (CTL is the temporal logic of Clarke and Emerson; trees model symbolic data structures such as lists). This can be used to infer types that help to detect programming errors wrt. a given temporal property. The results extend from CTL to the full modal μ-calculus thanks to their new results on the Horn μ-calculus. Observing a novel connection with so-called pushdown systems that model programs with recursive procedure calls, Charatonik and Podelski obtain new tests for pushdown systems and interprocedural data flow analysis through set constraint solving (see also the paragraph on constraint solving below).

Delzanno and Podelski call deductive model checking a new method that takes basically the same control structures (viz. fixpoint iterations) as model checking methods but is based on deduction (as opposed to exhaustive state space exploration). They employ the terminology and the formal concepts of Constraint Logic Programming (CLP) in order to show how one can perform program analysis by logical equivalence transformations on formulas (viz. CLP programs). The logical setup allows them to devise new optimizations (based on fixpoint evaluation strategies and on abstraction). They have implemented the method in a CLP system and use it experimentally to verify integer-valued protocols and parameterized systems and to detect array bound errors of C programs. The implementation shows a competitive performance on benchmark-like examples thanks to the built-in solver of arithmetic constraints.

Constraint Solving (Coordinators: Alexander Bockmayr, Andreas Podelski)

Under this headline we study both numerical constraints as they arise in mathematical optimization as well as constraints over symbolic structures (finite and infinite trees). Some of these logical formalisms (e.g., fragments and variants of second-order unification) are of general theoretical interest. Others, in particular set constraints, have direct applications to program analysis.

The group on constraint programming (Alexander Bockmayr, Thomas Kasper and Friedrich Eisenbrand) has continued and considerably extended its work at the interface of finite domain constraint programming and integer linear programming. They have developed a unifying logical framework, branch-and-infer, that clarifies the relationship between these two approaches and shows how they can be integrated. Moreover, they have studied the complexity of Gomory-Chvátal cutting planes. They proved a polynomial upper bound on the Chvátal rank of 0/1 polytopes and solved a longstanding open question by showing that the membership problem for the first elementary closure is coNP-complete.

Ganzinger, Jacquemard, and Veanes have shown that the symmetry of equality in rigid E-unification is crucial to some of the decidability and complexity results about this constraint domain. The non-symmetric variant, called rigid reachability, was shown to be undecidable already in the

single-constraint case, and, respectively, EXPTIME-complete in cases where simultaneous rigid E-unification is only P-complete. This has yielded a new undecidability result for a very restricted form of second-order unification. Some of this work is directly related to context unification, a restricted form of second-order unification that is relevant, for instance, for extensions of Knuth-Bendix completion to non-symmetric rewriting. The decidability of context unification has been open for about 10 years. The problem is very difficult as word unification appears as a special case of context unification. Vorobyov has looked at an extended case and shown that the $\forall\exists^5$-theory of context unification (the open case is the $\exists^*$-theory) is undecidable.

Charatonik and Podelski have continued to investigate subclasses of set constraints that are natural and useful for specialized program analyses. In particular, they have established the complexity for computing least, greatest and arbitrary fixpoint solutions; an extension of set constraints with μ-calculus operators leads to sophisticated forms of program analyses. They have found that there is an important gap: in many interesting cases, adding expressiveness to a subclass means pushing the time complexity from cubic immediately to EXPTIME. This tells us the drastic price of increasing precision in a corresponding program analysis.

Extended Modal Logic and Automated Theorem Proving (Coordinators: Harald Ganzinger, Andreas Nonnengart)

Research in this area has proceeded mainly along two lines. We have continued to study semantic embeddings of modal logics into first-order logic so as to be able to apply specific instances of standard theorem proving methods. Schmidt has investigated the so-called optimized functional translation method and has, in particular, shown how to use E-unification and E-resolution as decision procedures on the resulting first-order path logics. Nonnengart has found a method for simplifying modal frame axioms using auxiliary modalities.

Second, we have continued our investigations into more general logical formalisms (the guarded fragment of first-order logic, hybrid logics, labelled deduction systems) that are relevant as abstractions of modal logics and variants. The guarded fragment was proposed as a fragment of first-order logic with the aim to better capture than two-variable logic some of the nice properties of propositional modal logic. Ganzinger, Meyer, and Veanes, however, have shown that the guarded fragment is very sensitive to adding transitivity axioms for binary relations. Decidability is lost even in the two-variable case, but can be recaptured if transitive relations may only appear in guards. Basin, Matthews, and Viganò have presented a systematic investigation of the basic proof theoretic properties of labelled deduction systems, starting from propositional modal logics and progressively extending our framework to deal with quantification and generalized non-classical modalities such as relevant implication and negation. Tzakova has identified tractable tableau-based proof systems for various hybrid logics. Hybrid logics are extensions of modal logics by concepts of names for possible worlds.

Logic and Uncertainty (Coordinators: Manfred Jaeger, Emil Weydert)

In this area, our work has centered on three topics: deontic logic as a basis for qualitative decision theory, probabilistic foundations for default reasoning, and formal systems for quantitative probabilistic reasoning.

Van der Torre has investigated new preference-based and labelled logics for obligations and desires, where he has tried in particular to combine the qualitative, goal-oriented with the quantitative, decision-theoretic perspective.

Weydert has proposed and investigated a procedure that constructs for a given default knowledge base a canonical, intuitively most plausible, ranked model. The construction process can be

interpreted as semi-qualitative entropy maximization, and thereby establishes a new interesting link between purely qualitative default reasoning and quantitative probabilistic reasoning.

Jaeger has been able to extend the standard concept of Bayesian networks by a notion of random relations over arbitrary domains, thereby considerably extending the expressiveness of that formalism, and creating a new formal framework for the study of random relational structures.

3 The Reactive and Hybrid Systems Group

A *reactive* system is a digital process (software, hardware, or a combination thereof) which carries out an ongoing interaction with its environment. For example, each individual process of a distributed program is reactive; the environment is formed by the remaining processes of the program. Perhaps the key characteristic feature of reactive systems is *concurrency*: the system and the environment proceed concurrently. A *hybrid* system is a reactive system whose environment includes analog processes. For example, a digital controller with analog sensors and actuators is hybrid. Perhaps the key characteristic feature of hybrid systems is *real time*: the discrete system interacts with the continuous environment in real time.

As digital devices permeate our daily lives, reactive and hybrid systems become ubiquitous and are increasingly deployed also in safety-critical applications. Typical examples include the avionics systems which provide electronic support for flying an aircraft, and the air traffic control systems which coordinate the flight of multiple aircraft.

With respect to complexity management, current computer-aided design and integration tools for reactive and hybrid systems lag behind current technological capabilities in hardware manufacturing and software production. With design and integration methodologies lacking, testing is often the last and only resort for ensuring product quality. An overreliance on testing, however, exposes two major problems. First, testing is very expensive, because errors are found late in the system development and integration process, which, as a result, can be thrown back to the initial design phase. Second, testing is notoriously unreliable for concurrent and real-time processes, whose errors often depend on multiple irreproducible contingencies. Some devastating examples of failures that were not uncovered despite extensive testing include the hardware bug of the Pentium floating-point unit and the software bug that caused the Ariane 5 explosion.

Our goal is to develop methods and tools for the design and analysis of reactive and hybrid systems which can help to prevent and detect errors during the design phase. Our methods and tools are to be systematic —i.e., based on a formal scientific foundation— and scalable —i.e., applicable to systems of industrial complexity. The main focal points of our research correspond to the first three stages in the design of reactive and hybrid systems: modeling, validation, and implementation.

Reactive and hybrid modeling Formal models for reactive and hybrid systems permit the system designer to specify with mathematical precision both system prototypes and system requirements. Formal modeling is necessary for determining design flaws as early as possible. Our efforts are concentrated on the development and investigation of two formal models, one for reactive computation which is particularly suitable for highly heterogeneous designs, and the other for hybrid computation.

Reactive Modules [1] is a new formalism for the specification of reactive systems, with special emphasis on modular and hierarchical specification. Modularity permits the formal integration of heterogeneous system components, such as software and hardware components, real-time and speed-independent components, and synchronously and asynchronously interacting processes. Hierarchy permits abstract views of a system at various levels of spatial and temporal granularity. For example, a processor can be naturally viewed at the gate/cycle level or at the register-transfer/instruction level. Reactive Modules are supported by a computer-aided design, simulation, and verification toolkit called MOCHA [3]. The toolkit is currently being extended to exploit design structure in verification (see below).

Hybrid Automata [5] is one of the original formalisms for the specification of mixed discrete-continuous systems. The analysis of Hybrid Automata is supported by a toolkit called HyTech [6]. This toolkit includes the first automatic verification tool for hybrid systems, which is based on symbolic model checking for certain classes of Hybrid Automata. As the main emphasis of HyTech concerns the modeling and analysis of hybrid behavior, the formalism of Hybrid Automata does not support orthogonal issues such as spatial and temporal structuring mechanisms. We are currently making use of our experiences with Reactive Modules and Hybrid Automata by developing a combined formalism of *Hybrid Modules*, which permits the modular and hierarchical specification and verification of hybrid systems. We are also extending HyTech to a new tool, called HyPerTech [7], which employs interval numerical methods for the analysis of hybrid systems with continuous dynamics that are considerably more general than what can be solved by purely symbolic methods.

Model checking and synthesis Model checking is an algorithmic technology for automatically checking if a formal model of a design meets a formal requirement specification. Model checking is successfully used in hardware design, but the application of model checking to heterogeneous, hierarchical systems with several hardware and software layers is still in its infancy. In large part, this is due to fact that the standard model-checking algorithms work on flat, unstructured state-transition models of designs. To remedy this situation, we concentrate on exploiting design structure in model checking. Design structure arises both in component-based and in hierarchical design. We are developing a model-checking methodology that makes use of both kinds of design structure.

Compositional model checking applies a divide-and-conquer approach to the verification problem. In order to decompose a verification task for a compound system into subtasks for the components, it is usually necessary to consider each individual component not in isolation but relative to certain assumptions about the context. This approach is often called assume-guarantee decomposition. *Hierarchical model checking* applies multiple levels and facets of design abstraction in order decrease the distance between a design and its requirements and in order to orthogonalize different aspects of the verification, such as functionality and timing. We hope to achieve a qualitative breakthrough in the applicability of model checking by integrating compositional and hierarchical techniques. For example, using assume-guarantee decomposition while preserving multiple time lines we have detected fully automatically several subtle design errors in a 100-processor video-graphics-image chip, which is far beyond the scope of current model-checking tools [8].

The components of a system and its environment can be viewed as players in a game whose objective is to either satisfy (in the case of the system) or violate (in the case of the environment) the desired system requirements. The study of such games leads not only to compositional techniques in model checking, but often enables the automatic derivation of winning or spoiling strategies as well. This approach, called *model synthesis*, is particularly promising in control, where it allows the automatic derivation of digital controllers for reactive and hybrid systems. In practice, model synthesis has been applied only in game settings where the individual players pursue simple safety objectives by applying deterministic strategies. Our work is focused on the efficient and also the probabilistic solution of more general game objectives, and on the development of specification formalisms for game requirements. For example, the temporal-logic based specification formalism Atl permits the efficient synthesis of both deterministic and randomized models (controllers) [2, 4].

Reactive programming and real-time implementation Once an embedded design is validated, it needs to be partitioned into hardware and software and realized on a given resource

platform. Resource restrictions may involve diverse measures such as cost or power consumption, and they may mandate the use of off-the-shelf components such as a particular piece of hardware, a particular scheduling algorithm, or a particular real-time operating system. In the short term, we will develop and implement a synchronous programming language on top of hybrid modules. This will enable us to simulate and prototype embedded designs under resource restrictions. In the medium term, we plan to fully equip an embedded systems lab. This will enable us to carry out concrete experiments in all phases of the design process, from modeling to implementation.

Current projects

MOCHA a design and verification platform that exploits modularity and hierarchy in model checking [3]

HYPERTECH a toolkit for modeling, simulating, and validating hybrid systems using interval numerical methods [7]

ATL a framework for specifying, analyzing, and synthesizing both collaborative and adversarial behavior in component-based designs [2]

References

[1] R. Alur and T.A. Henzinger. Reactive modules. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 207–218. IEEE Computer Society Press, 1996.

[2] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 100–109. IEEE Computer Society Press, 1997.

[3] R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S.K. Rajamani, and S. Tasiran. MOCHA: modularity in model checking. In A.J. Hu and M.Y. Vardi, editors, *CAV 98: Computer-aided Verification*, Lecture Notes in Computer Science 1427, pages 521–525. Springer-Verlag, 1998.

[4] L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 564–575. IEEE Computer Society Press, 1998.

[5] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[6] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.

[7] T.A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HYTECH: using interval numerical methods for hybrid systems analysis. In *Proceedings of the AAAI Spring Symposium at Stanford University*. AAAI, 1999.

[8] T.A. Henzinger, S. Qadeer, and S.K. Rajamani. Assume-guarantee refinement between different time scales. In N. Halbwachs and D. Peled, editors, *CAV 99: Computer-aided Verification*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

4 The Computer Graphics Group

During the last decades computer graphics has established itself as a core technique within computer science and information technology. Computer systems are more and more used to realize and simulate parts of the real or an imaginary world. Such simulations require to model, to render, and to animate complex objects. The goal of computer graphics is to turn abstract information into visual images and to allow the user to interact with complex objects and data in a natural and intuitive way.

In order to achieve these goals, computer graphics requires techniques for turning abstract data into a suitable computer model (modeling) and for converting computer models into images (rendering). Both techniques are also fundamental for the interactive exploration of complex structures in applications such as engineering or medical imaging (visualization, virtual reality).

The newly established computer graphics group at the Max-Planck-Institut für Informatik continues research previously carried out at the University of Erlangen and conducts basic research in the above mentioned areas. Current research focuses on the following topics:

- Freeform Curves and Surfaces
- Efficient Polygonal Meshes
- Simulation of Global Illumination in Complex Environments
- Image Based and Hardware Accelerated Photorealistic Rendering
- Visualization of Complex Medical and Engineering Data

Our work is both theoretical and practical with a focus on the development of novel algorithms, the integration of new algorithms into a system, and the evaluation of the system in practical applications.

We briefly survey some of the highlights in our main research areas:

Surface Reconstruction We have been working on the development of algorithms and on a system infrastructure for surface reconstruction from discrete data. Besides approximating the given data within a specified tolerance the system should generate smooth surfaces in a CAD-compatible format. Moreover, the system should be able to process complex shapes and handle large data sets.

We have developed algorithms for deriving good parametrizations of the data points, and for semi-automatic patch-layouting. The parametrization is obtained by interpreting the triangulated data as a mass-spring model. Patch layouting is based on a segmentation using discrete curvature computations, followed by the application of morphological operators.

Once the patch layout has been determined, actual surface reconstruction can be done by combining hierarchical refinement with variational or discrete fairing.

Mesh Reduction In many areas of computer graphics and computer aided geometric design, triangle meshes have become the standard for representing surface geometry. Triangle meshes arise naturally as output of laser range scanners, in medical imaging, or as output of mathematical simulations. Another source of very large triangle meshes is through the conversion of freeform surfaces.

The main drawback of triangle meshes is the large amount of data needed to represent a smooth nonplanar surface. This has inspired a lot of work in the field of mesh simplification, i.e., the reduction of the complexity of a triangle mesh, while maintaining a close approximation to the original model.

We have developed an algorithm for mesh decimation that is based on a suitable distance measure (one sided Hausdorff distance), a simple topological operator (half edge collapse), and a fairness predicate. The algorithm is fast and is guaranteed to produce an approximation to the original mesh within a user specified error tolerance. The choice of guidance predicate offers additional control over the quality of the resulting surface. The algorithm can be used efficiently for incremental mesh decimation and allows to convert an arbitrary triangle mesh into a progressive mesh representation. The algorithm also allows to handle attributes such as color.

Interactive Multiresolution Modeling on Arbitrary Triangle Meshes We have generalized powerful multiresolution techniques from subdivision surfaces to arbitrary triangle meshes without requiring subdivision connectivity. Our major observation is that the hierarchy of nested spaces which is the structural core element of most multiresolution algorithms can be replaced by the sequence of intermediate meshes emerging from the application of incremental mesh decimation. Performing such schemes with local frame coding of the detail coefficients already provides effective and efficient algorithms to extract multiresolution information from unstructured meshes.

In combination with discrete fairing techniques, i.e., the constrained minimization of discrete energy functionals, we have obtain fast mesh smoothing algorithms which are able to reduce noise from a geometrically specified frequency band in a multiresolution decomposition. Putting mesh hierarchies, local frame coding and multi-level smoothing together has allowed us to propose a flexible and intuitive paradigm for interactive detail-preserving mesh modification.

Global Illumination Computations Hierarchical Radiosity (HR) and its extensions towards clustering have established themselves as the standard finite-element method for global illumination computations in diffuse environments. Nevertheless, for complex scenes HR is often not applicable because of its enormous memory requirements. The major memory consumption arises from the storage of the interaction coefficients - or links - between the interacting objects in the scene. This storage is necessary due to the gathering scheme that is usually used in the context of HR.

Applying another iterative method, namely Southwell Relaxation (in the context of radiosity often referred to as shooting) this storage of links can be avoided with a tolerable increase in computation time and solution error.

Measurements for some example scenes show that the error behavior is slightly worse than for gathering, but visually no difference can be seen between the two solutions. Finally, a link caching scheme was examined, where links which are deemed most probable to be reused in the next iteration, are cached and thus do not have to be recomputed later on. This allows to define a certain memory budget to be spent on link storage. With the mentioned improvements our system is able to handle scenes consisting of over half a million patches.

Other improvements include uniform handling of curved surfaces and extensions of the algorithm to non-diffuse environments.

Adaptive Lumigraph Acquisition from Synthetic Scenes and Canned Light Sources Light fields and Lumigraphs are capable of rendering scenes of arbitrary geometrical or illumination complexity in real time. They are thus interesting ways of interacting with both recorded real-world and high-quality synthetic scenes.

Unfortunately, both light fields and Lumigraph rely on a dense sampling of the illumination to provide a good rendering quality. This induces high costs both in terms of storage requirements and computational resources for the image acquisition. Techniques for acquiring adaptive light field and Lumigraph representations are thus mandatory for practical applications.

We have developed an algorithm for the adaptive acquisition of images for Lumigraphs from synthetic scenes. Using image warping to predict the potential improvement in image quality when adding a certain view, we decide which new views of the scene should be rendered and added to the light field. This a-priori error estimator accounts for both visibility problems and illumination effects such as specular highlights.

Another application of light fields is the inclusion of realistic light sources in image synthesis. For a given lamp geometry and luminary, the outgoing lightfield is computed using standard global illumination methods, and stored away in a Lumigraph data structure. Later the lightfield can be used to illuminate a given scene while abstracting from the original lamp geometry. We call a light source stored and used in this fashion a Canned Lightsource.

Realistic, Hardware-accelerated Shading and Lighting With fast 3D graphics becoming more and more available on low end platforms, the focus in hardware-accelerated rendering is beginning to shift towards higher quality rendering and additional functionality instead of simply higher performance implementations based on the traditional graphics pipeline.

We have developed novel algorithms and techniques for realistic shading and lighting using computer graphics hardware. In particular, we have developed an algorithm for high-quality local illumination using physically plausible lighting models (as, e.g. Torrance Sparrow). The approach is based on an analytic factorization of the respective model into bivariate terms that can be represented as texture maps.

We have also developed methods for visualizing non-diffuse global illumination solutions based on environment maps. We introduce both a Fresnel term for simulating reflections in non-metallic objects, as well as a pre-filtering method for environment maps. We have also developed an alternative parametrization for environment maps that allows us to use one map for all viewing positions and directions. These techniques can finally be combined with normal mapping to increase the visual complexity of the scene.

Real-Time Exploration of Regular Volume Data by Adaptive Reconstruction of Iso-Surfaces Recent advances in the technology of 3D sensors and in the performance of numerical simulations result in the generation of volume data of ever growing size. In order to allow real-time exploration of even the highest resolution data sets, adaptive techniques benefiting from the hierarchical nature of multiresolution representations have gained special attention.

We have developed an adaptive approach to the fast reconstruction of iso-surfaces from regular volume data at arbitrary levels of detail. The algorithm has been designed to enable real-time navigation through complex structures while providing user-adjustable resolution levels. Since adaptive on-the-fly reconstruction and rendering is performed from a hierarchical octree representation of the volume data, the method does not depend on pre-processing with respect to a specific iso-value, thus allowing the user to interactively browse through the pencil of iso-surfaces. Special attention has been paid to the fixing of cracks in the surface where the adaptive reconstruction level changes and to the efficient estimation of the iso-surface's curvature.

Part III

The Algorithms and Complexity Group

1 Personnel

Director:
Prof. Dr. Kurt Mehlhorn

Research Associates:
Dr. Susanne Albers
Dr. Christoph Burnikel
Dr. Shiva Chaudhuri (until November 97; now at Synopsys Inc., Mountain View, California, USA)
Dr. Rudolf Fleischer
Dr. Naveen Garg (until December 97; now at Indian Institute of Technology, New Delhi, India)
Prof. Dr. Torben Hagerup (until June 98; now at Universität Frankfurt, Germany)
Dr. Hans-Peter Lenhof
Dr. Petra Mutzel
Dr. Edgar Ramos
Dr. Christine Rüb
Dr. Stefan Schirra
Dr. Jop F. Sibeyn
Dr. Christian Uhrig
Dr. Christos Zaroliagis (until August 98; now at King's College, University of London, United Kingdom)

Postdocs:
Dr. Gerth Stølting Brodal (until July 98; now at Datalogisk Institut, Århus Universitet, Denmark)
Dr. Ka-Wong Chong (until August 98; now at University of Hong Kong)
Dr. Paolo Ferragina (since September 97, until September 98; previously and now at Unversity of Pisa, Italy)
Dr. Xudong Fu (since September 97; previously at McGill University, Montreal, Canada)
Dr. Leszek Gasieniec (until October 97; now at University of Liverpool, United Kingdom)
Dr. Susan Hert (since January 99; previously at Knox College, Galesburg, USA)
Dr. Klaus Jansen (since September 97; previously at Universität Trier, Germany; until April 98; now at IDSIA, Lugano, Switzerland)
Dr. Zeev Nutov (since August 98; previously at University of Waterloo, Canada)
Dr. Marina Papatriantafilou (until September 97; now at Chalmers University of Technology, Goeteborg, Sweden)
Dr. Lorant Porkolab (since August 97; previously at RUTCOR, Rutgers University, New Brunswick, USA)
Dr. Peter Sanders (since October 97; previously at Universität Karlsruhe, Germany)
Dr. Steven Seiden (since September 98; previously at Technische Universität Graz, Austria)
Dr. Roberto Solis-Oba (since September 97; previously at Purdue University, West Lafayette, USA)
Dr. C.R. Subramanian (since January 98; previously at Indian Institute of Technology, Panbazar, Guwahati, India)
Dr. Philippas Tsigas (until September 97; now at Chalmers University of Technology, Goeteborg, Sweden)

Ph.D. students:
Ernst Althaus (since February 99)
Hannah Bast
Stefan Burkhardt (since November 97)
Andreas Crauser

Stefan Funke (since April 1998)
Jordan Gergov
Christoph Hundack
Gunnar Klau (since July 97)
Oliver Kohlbacher
Piotr Krysta (since October 1997)
Ulrich Meyer
Peter Müller
Volker Priebe
Knut Reinert (until April 99)
Michael Seel
René Weiskircher (since September 97)
Mark Ziegelmann (since April 98)
Joachim Ziegler (since December 97)
Thomas Ziegler

Secretaries:
Ingrid Finkler
Evelyn Haak
Martina Horn

2 Data Structures and Graph Algorithms

Combinatorial algorithms continue to be an active area of research in our group spanning a wide range of topics. Over the past two years we have concentrated on the following:

- elementary data structures
- graph and network algorithms
- implementation of algorithms.

Our work is still mainly of a theoretical nature, but with a growing awareness of the importance of really implementing algorithms and studying their behavior experimentally rather than by mathematical analysis alone.

Within *Elementary Data Structures* (Section 2.1), we have studied several problems concerning the efficient storage and retrieval of data in various models of computation. For comparator networks, we showed how to build heaps with networks of optimal size and depth. In the pointer machine model, we presented the first finger search tree which allows insertion of new elements in worst case constant time (deletions take time $O(\log^* n)$). For the word RAM model, we improved previous upper and lower bounds for maintaining static dictionaries.

Another line of research focused on the efficient implementation of certain operations in data structures for database systems with concurrent accesses. We proposed a new group update scheme for relaxed height-balanced search trees. And we showed how to efficiently maintain a full-text index in constantly changing environments (such as WWW pages) where failures are not uncommon.

Within *Graph and Network Algorithms* (Section 2.2), our research centered around computing maximum flows, minimum cuts, and shortest paths. We gave a simpler proof, based on a potential function, for the $O(n^2\sqrt{m})$ running time of Goldberg and Tarjan's preflow-push maximum flow algorithm with highest-level selection rule. We also proposed several variants of Goldberg and Rao's binary blocking flow algorithm, which seem to be superior to the original algorithm, at least in our experiments (however, they can still not compete with preflow-push algorithms). For k-terminal networks, we continued our study of mimicking networks; in particular, we improved the upper bound on the minimum size of mimicking networks for networks of bounded treewidth from exponential to linear in k. We extended the Stoer-Wagner min-cut algorithm in weighted undirected graphs such that it also computes in linear time a maximum flow. And we showed how to represent minimum and sub-minimum cuts with $O(n)$ space using a special type of cactus-trees.

Besides investigating flow problems, we continued to study the average-case complexity of the all-pairs shortest-path problem on directed graphs. In the $G(n,p)$ model and in the vertex potential model, our algorithms have running time $O(n^2 \log n)$ with high probability. The same bound holds for graphs with random integer weights. We also started to work on the k-resource constrained shortest-path problem for which we developed methods that sometimes solve a relaxed LP formulation.

In another direction of research, we showed a tight bound of $3k-1$ for the width of a tree decomposition of graphs with treewidth k if the diameter of the trees is restricted to $O(\log n)$. For random k-colorable graphs we gave a simple BFS-based algorithm for k-coloring, which is efficient with high probability. Matroid theory provides a general framework for network optimization problems; we studied the robustness of matroid optimization problems with respect to perturbations of the inputs.

Within *Implementing Algorithms* (Section 2.3), we discussed the question of whether actually implementing algorithms is beneficial for theoretical computer scientists. We did some extensive experiments with dynamic algorithms for directed graphs, and — on a smaller scale — with max-flow algorithms and algorithms for constrained shortest-path problems. We continued to implement algorithms in our library of online algorithms and started a new project aimed at animating elementary data structures and basic algorithms (as they can be found in LEDA, for example).

2.1 Elementary Data Structures

Comparator Networks for Binary Heap Construction

Investigator: Gerth Stølting Brodal

The heap data structure, introduced in 1964 by Williams [6], has been extensively investigated in the literature due to its many applications and intriguing partial order. Algorithms for heap management—insertion, minimum deletion, and construction—have been discussed in several models of computation. Floyd [4] has given a sequential algorithm for building a heap in a bottom-up fashion in linear time, which is clearly optimal. On the EREW PRAM Olariu and Wen [5] can build a heap of size n in time $O(\log n)$. For the CREW PRAM model a heap construction algorithm was given by Raman and Dietz [3] that takes $O(\log\log n)$ time. The same time performance holds for the parallel comparison tree model [2]. Finally Dietz showed that $O(\alpha(n))$, where $\alpha(n)$ is the inverse of Ackerman's function, is the expected time required to build a heap in the randomized parallel comparison tree model [2]. All the above parallel algorithms achieve optimal work $O(n)$, and the time optimality of the deterministic algorithms can be argued by reduction from the selection of the minimum element in a set.

In [1], we considered the heap construction problem for the simplest parallel model of computation, namely *comparator networks*. We showed that heap construction can be done by comparator networks of size $O(n\log\log n)$ and depth $O(\log n)$, and that our networks reach optimal size by reducing the problem of selecting the smallest $\log n$ elements to heap construction. Since finding the minimum requires at least a network of size $n-1$ and depth $\lceil\log n\rceil$ our heap construction networks have optimal depth.

References

- [1] Gerth Stølting Brodal and M. Cristina Pinotti. Comparator networks for binary heap construction. In *Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (SWAT'98)*, volume 1432 of *Lecture Notes in Computer Science*, pages 158–168. Springer, Berlin, 1998. To appear in Theoretical Computer Science.

[2] P. F. Dietz. Heap construction in the parallel comparison tree model. In *Proceedings of the 3rd Scandinavian Workshop on Algorithm Theory (SWAT'98)*, volume 621 of *Lecture Notes in Computer Science*, pages 140–150. Springer, Berlin, 1992.

[3] P. F. Dietz and R. Raman. Very fast optimal parallel algorithms for heap construction. In *Proceedings of the 6th Symposium on Parallel and Distributed Processing (SPDP'94)*, pages 514–521, 1994.

[4] R. W. Floyd. Algorithm 245: Treesort3. *Communications of the ACM*, 7(12):701, 1964.

[5] S. Olariu and Z. Wen. Optimal parallel initialization algorithms for a class of priority queues. *IEEE Transactions on Parallel and Distributed Systems*, 2:423–429, 1991.

[6] J. William J. Williams. Algorithm 232: Heapsort. *Communications of the ACM*, 7(6):347–348, 1964.

Finger Search Trees with Constant Update Time

Investigator: Gerth Stølting Brodal

A finger search tree is a data structure that stores a sorted list of elements in such a way that searches are fast in the vicinity of a *finger*, where a finger is a pointer to an arbitrary element of the list.

Brown and Tarjan [2] observed that by *level-linking* $(2,4)$–trees, finger searches can be done in worst case $O(\log \delta)$ time, where δ is the difference between the ranks of the finger and the search element in the list. In the following, we denote a data structure having $O(\log \delta)$ search time a finger search tree. Huddleston and Mehlhorn [7] showed that $(2,4)$–trees support insertions and deletions in amortized constant time, assuming that the position of the element to be inserted or deleted is known.

The question we considered in [1] is, whether it is possible to remove the amortization from the result of Huddleston and Mehlhorn [7], *i.e.*, if finger search trees exist that support insertions and deletions in worst case constant time.

By assuming a unit-cost RAM, Dietz and Raman [3] presented a finger search tree implementation supporting insertions and deletions in worst case constant time. Their data structure is based on the standard RAM technique of packing small problem sizes into a constant number of machine words. For the weaker pointer machine model no similar result is known. For the pointer machine, finger search trees that obtain worst case $O(\log^* n)$ insertion and deletion time have been given by Harel and Lueker [5, 6], where n is the length of the list. Search trees with constant insertion and deletion time on the pointer machine have been presented by Levcopoulos and Overmars [8] and Fleischer [4], but neither of them supports finger searches.

In [1], we presented the first finger search tree implementation for the pointer machine that supports finger searches and which supports insertions in worst case constant time. The data structure supports deletions in worst case $O(\log^* n)$ time, which matches the previous best bounds of Harel and Lueker [5, 6]. The space requirement for the data structure is $O(n)$.

References

- [1] Gerth Stølting Brodal. Finger search trees with constant update time. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 540–549, 1998.

[2] M. R. Brown and R. E. Tarjan. Design and analysis of a data structure for representing sorted lists. *SIAM Journal of Computing*, 9:594–614, 1980.

[3] P. F. Dietz and R. Raman. A constant update time finger search tree. *Information Processing Letters*, 52:147–154, 1994.

[4] Rudolf Fleischer. A simple balanced search tree with $O(1)$ worst-case update time. *International Journal of Foundations of Computer Science*, 7:137–149, 1996.

[5] Dov Harel. Fast updates of balanced search trees with a guaranteed time bound per update. Technical Report 154, University of California, Irvine, 1980.

[6] Dov Harel and George S. Lueker. A data structure with movable fingers and deletions. Technical Report 145, University of California, Irvine, 1979.

[7] Scott Huddleston and Kurt Mehlhorn. A new data structure for representing sorted lists. *Acta Informatica*, 17:157–184, 1982.

[8] C. Levcopoulos and M. H. Overmars. A balanced search tree with $O(1)$ worst-case update time. *Acta Informatica*, 26:269–277, 1988.

Algorithms and Data Structures for the Word RAM

Investigator: Torben Hagerup

A *word RAM* is a unit-cost random-access machine with a word length of w bits, for some positive integer w, and with an instruction repertoire similar to that found in present-day computers. In the previous progress report we described an algorithm that sorts n numbers (integers in the range $\{0, \ldots, 2^w - 1\}$) in $O(n \log\log n)$ time. That result was part of the beginning of an intensive study by us and by others of the capabilities of the word RAM, one high point of which was the demonstration by Thorup [5] that the single-source shortest-paths problem can be solved in linear time for undirected networks (again, all edge weights are assumed to be integers in the range $\{0, \ldots, 2^w - 1\}$). Although algorithms and data structures for the word RAM are still a very active research area that may hold many surprises in store, some parts of that area appear to have come of age, and in [4] we gave a careful survey of what is known about sorting and searching on the word RAM. Putting together all the known facts in a systematic way yielded some additional benefits in the form of new results. For example, we showed for the first time that if w is very large relative to n, deterministic sorting in linear time and space is possible on the word RAM.

In [3] we considered a specific data-structuring problem for the word RAM, namely that of providing a static dictionary for n w-bit keys with associated satellite information. A *static dictionary* for a set of keys is a data structure that admits only one kind of operation, namely *lookup* queries: Given a key x, is it one of the keys stored in the data structure? If so, what is its associated satellite information? Parameters of interest for a given static dictionary are the *query time* and the *space* needed to store n keys. In a celebrated result, Fredman *et al.* [2] showed that one can achieve all that one could have hoped for, constant query time together with a linear space bound. The data structure of Fredman *et al.* depends intimately on the availability of unit-time multiplication, an assumption that is eyed with suspicion by some because multiplication does not belong to the complexity class AC^0, *i.e.*, cannot be realized in constant time by circuits of polynomial size (polynomial in the word length w, that is). For this reason, Andersson *et al.* asked in [1] what can be done if only AC^0 instructions are allowed and provided upper and lower bounds for the achievable query time under the assumption that only $O(n)$ space is available for the data structure. In particular, a constant query time is not possible for all combinations of n and w, in contrast with what is the case if unit-time multiplication is available. Building strongly on the work of Andersson *et al.*, we gave a cleaner presentation of the core ideas and provided strengthened upper and lower bounds. In particular, if $w/\log n = (\log\log n)^{O(1)}$, *i.e.*, if w is larger than $\log n$ (as it must be for the problem to make sense), but not much larger, then query time $O(\log\log\log n)$ is possible—Andersson *et al.* achieved query time $O(\log\log n)$ for this case.

References

[1] A. Andersson, P. B. Miltersen, S. Riis, and M. Thorup. Static dictionaries on AC^0 RAMs: Query time $\Theta(\sqrt{\log n/\log\log n})$ is necessary and sufficient. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS'96)*, pages 441–450, 1996.

[2] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.

- [3] T. Hagerup. Simpler and faster dictionaries on the AC^0 RAM. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP'98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 79–90. Springer, Berlin, 1998.

- [4] T. Hagerup. Sorting and searching on the word RAM. In *Proceedings of the 15th Symposium on*

Theoretical Aspects of Computer Science (STACS'98), volume 1373 of *Lecture Notes in Computer Science*, pages 366–398. Springer, Berlin, 1998.

[5] M. Thorup. Undirected single source shortest paths in linear time. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS'97)*, pages 12–21, 1997.

Group Updates for Relaxed Height-Balanced Trees

Investigator: Eljas Soisalon-Soininen

A group update of a search tree means that a set of insertions or deletions are collected into a group in sorted order, and this group is brought into the tree as a single transaction (see [6], *e.g.*). In [3], we presented an efficient group-update algorithm for height-balanced binary search trees. The algorithm is based on relaxed balancing [2, 5, 4] and it has two steps: First, the operations in the underlying group are performed without any balancing except for subgroups between two consecutive keys in the original tree. In this way the updates are made available as soon as possible without sacrificing the logarithmic search time. In the second step the tree is balanced, *i.e.*, transformed into a tree satisfying the (local) balance criteria of height-balanced trees. Balancing is designed as a background process allowing the concurrent use of the structure. The balancing time is comparable to earlier results [1] in cases when balancing is strictly connected with individual updates.

References

[1] M. R. Brown and R. Endre Tarjan. Design and analysis of a data structure for representing sorted lists. *SIAM Journal of Computing*, 9:594–614, 1980.

[2] K. S. Larsen, E. Soisalon-Soininen, and P. Widmayer. Relaxed balance through standard rotations. In *5th International Workshop on Algorithms and Data structures (WADS'97)*, volume 1272 of *Lecture Notes in Computer Science*, pages 450–461. Springer, 1997.

- [3] L. Malmi and E. Soisalon-Soininen. Group updates for relaxed height-balanced trees. In *Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1999. To appear.

[4] O. Nurmi and E. Soisalon-Soininen. Chromatic binary search trees: a structure for concurrent rebalancing. *Acta Informatica*, 33:547–557, 1996.

[5] O. Nurmi, E. Soisalon-Soininen, and D. Wood. Concurrency control in database structures with relaxed balance. In *Proceedings of the 6th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (SPDP'87)*, pages 170–176, 1987.

[6] K. Pollari-Malmi, E. Soisalon-Soininen, and T. Ylönen. Concurrency control in B-trees with batch updates. *IEEE Transactions on Knowledge and Data Engineering*, 8:975–984, 1996.

Concurrency and Recovery in Full-Text Indexing

Investigator: Eljas Soisalon-Soininen

An important feature of a document database system is that the documents can be retrieved by searching for words from their contents. In a full-text index [1, 2, 3], each word of the stored documents can be used as a search key. Inserting a new document into the database automatically triggers a transaction that inserts the words together with their occurrence information into the index. In [8], we gave solutions to problems that arise when full-text indexing is applied for constantly changing document data, such as WWW pages. In particular, we presented an algorithm for

full-text indexing with the following properties: Concurrent searches are possible and efficient, and the algorithm can be designed such that several indexing processes can be performed concurrently. Moreover, the algorithm allows efficient recovery of the index after failures that can occur while the index is modified. This is important for large indices, because when not prepared for failures, the index may need to be reconstructed from original documents.

The recovery method of [8] is based on path copying reminiscent to shadow paging [4, 6]. Other index recovery methods, based on logging, have been presented in [7, 5].

References

[1] V. N. Anh and A. Moffat. Compressed inverted files with reduced decoding overheads. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 290–297, 1998.

[2] A. F. Cardenas. Analysis and performance of inverted data base structures. *Communications of the ACM*, 18:695–702, 1975.

[3] D. Cutting and J. Pedersen. Optimization for dynamic inverted index maintenance. In *Proceedings of the 13th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405–411, 1990.

[4] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann Publishers, San Mateo, Ca., 1993.

[5] M. Kornacker, C. Mohan, and J. M. Hellerstein. Concurrency and recovery in generalized search trees. In *Proceedings of the ACM SIGMOD Conference*, 1997.

[6] V. Kumar and M. Hsu, editors. *Recovery Mechanisms in Database Systems.* Prentice Hall, NJ, 1998.

[7] D. Lomet and B. Salzberg. Concurrency and recovery for index trees. *VLDB Journal*, 6:224–240, 1997.

- [8] E. Soisalon-Soininen and P. Widmayer. Concurrency and recovery in full-text indexing, 1999. Manuscript.

2.2 Graph and Network Algorithms

The Max-Flow Problem

Investigators: Torben Hagerup, Kurt Mehlhorn, and Peter Sanders

Despite intensive research for more than three decades, problems related to flows in networks still motivate cutting-edge algorithmic research. Goldberg and Tarjan [4] introduced the preflow-push method for solving this problem. When this method is implemented with the highest-level selection rule, then both the running time and the number of pushes are known to be $O(n^2\sqrt{m})$, where n is the number of nodes and m is the number of edges. In [1], we gave a new proof based on a potential function argument. Potential function arguments may be preferable for analyzing preflow-push algorithms, since they are simple and generic.

Recently, Goldberg and Rao [3] proposed the *binary blocking flow* (BBF) algorithm for computing a maximum flow in a flow network with n nodes and m edges with integer capacities bounded by U in time $O(m\Lambda \log(n^2/m) \log U)$, where $\Lambda = \min\{m^{1/2}, n^{2/3}\}$. This is significantly smaller than the long-standing bound of $O(n^3/\log n)$. It is therefore a natural question whether the new algorithm might have a similar impact on the performance of actual implementations as the development of preflow push algorithms had [2].

In order to investigate this question, we designed and analyzed a more general family of BBF algorithms that have the same worst case behavior as the original algorithm and contain it as a

special case [5]. In addition, some variants promise better performance in practice. In particular, the algorithm by Goldberg and Rao builds acyclic networks by contracting strongly connected components of the network into single nodes, which are guaranteed to be able to carry a certain flow Δ. It then computes a blocking flow on the resulting acyclic network. In order to be able to transfer this blocking flow into an augmenting flow in the original network, all flow beyond Δ is returned to the source. In contrast, our technique replaces strongly connected components by acyclic networks with the property that any blocking flow computed on them can easily be transferred into an augmenting flow for the original network.

We implemented several variants of this approach in `C++` using the LEDA library and employed the test instances and algorithms used in a previous study [2] to assess the performance of our approach [5]. It seems to be faster than the original BBF algorithm. This advantage can be further amplified using heuristics to increase the capacity of the acyclic networks generated by the algorithm. However, so far the BBF algorithm can only compete with highly developed implementations of the preflow push technique for rather few classes of inputs. From the experiments performed so far, it is hard to predict whether the BBF algorithm will eventually emerge as the fastest algorithm for practical inputs. There are too many untried potentially useful heuristics and experience with preflow push algorithms suggests that heuristics can have a large impact on practical performance.

References

- [1] J. Cheriyan and K. Mehlhorn. An analysis of the highest-level selection rule in the preflow-push max-flow algorithm. *Information Processing Letters*, 69(5):239–242, 1999.

  [2] B. V. Cherkassky and A. V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1997.

  [3] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):1–15, 1998.

  [4] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *Journal of the ACM*, 35:921–940, 1988.

- [5] T. Hagerup, P. Sanders, and J. L. Träff. An implementation of the binary blocking flow algorithm. In K. Mehlhorn, editor, *2nd Workshop on Algorithm Engineering (WAE'98)*, number MPI-I-98-1-019, ISSN: 0946-011X in Research Reports MPII, pages 143–154, 1998.

Mimicking Networks

Investigators: Shiva Chaudhuri and Christos Zaroliagis

One of the central (and classical) problems in network flows is the characterization of the flow behavior of multi-terminal networks, *i.e.*, networks with $k > 2$ terminals, first motivated and solved by Gomory and Hu and later improved and simplified by many others. The Gomory-Hu approach, as well as its subsequent improvements and simplifications, deals mainly with the case where every vertex of the network is a potential candidate for being a terminal. However, there may be cases where the number of terminals is much smaller than the number of vertices in the network.

Under this perspective, there is a recent, renewed interest on the problem of characterizing the flow behavior of networks with a small (usually constant) number of terminals [1, 3]. More precisely, Hagerup *et al.* showed [3] that for any k-terminal network G there exists a network $M(G)$, called a *mimicking network*, with 2^{2^k} vertices — k of which are terminals — that has the

same feasible external flows as G. If, in addition, the input network is outerplanar, then Arikati *et al.* showed [1] that there exists a better mimicking network of size $k^2 2^{k+2}$, which is also outerplanar. Mimicking networks constituted the main building block in the development of optimal algorithms for computing a maximum s-t flow [3] and all-pairs min-cut [1] in a bounded treewidth network as well as of improved algorithms for computing an s-t min-cut and all-pairs min-cut in planar networks [1].

A natural question is whether there are more efficient constructions of mimicking networks, *i.e.*, constructions such that $|M(G)|$ does not depend single- or double-exponentially on k. We have recently made a step forward in answering this question [2]. Let $S(k)$ denote the minimum size of a mimicking network for a k-terminal network. We have proved the following results (the values in brackets are the previously best known results): $S(4) = 5$ $[2^{16}]$, $S(5) = 6$ $[2^{32}]$. For bounded treewidth networks we showed $S(k) = O(k)$ $[2^{2^k}]$, and for outerplanar networks we showed $S(k) \leq 10k - 6$ $[k^2 2^{k+2}]$.

References

[1] S. Arikati, S. Chaudhuri, and C. Zaroliagis. All-pairs min-cut in sparse networks. *Journal of Algorithms*, 29:82–110, 1998.

- [2] S. Chaudhuri, K. V. Subrahmanyam, F. Wagner, and C. Zaroliagis. Computing mimicking networks. In *Proceedings of the 25th International Colloqium on Automata, Languages and Programming (ICALP'98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 556–567. Springer, 1998. To appear in *Algorithmica*, Special Issue on Graph Algorithms and Applications.

[3] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde. Characterizations of k-terminal flow networks and computing network flows in partial k-trees. *Journal of Computer and System Sciences*, 57:366–375, 1998.

The Min-Cut Problem

Investigator: Kurt Mehlhorn

The Stoer-Wagner algorithm [2] computes a minimum cut in a weighted undirected graph G. The algorithm works in $n - 1$ phases, where n is the number of nodes of G. Each phase takes time $O(m + n \log n)$, where m is the number of edges of G, and computes a pair of vertices s and t and a minimum cut separating s and t. In [1], we showed how to extend the algorithm such that each phase also computes a maximum flow from s to t. The flow is computed in $O(m)$ additional time and certifies the cut computed in that phase.

References

- [1] S. Arikati and K. Mehlhorn. A correctness certificate for the Stoer-Wagner mincut algorithm. Manuscript, see `www.mpi-sb.mpg.de/~mehlhorn/ftp/mincut.ps`, 1999.

[2] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

Subminimum Edge Cuts of a Weighted Graph

Investigator: Zeev Nutov

Let λ denote the minimum weight of an edge cut of a graph $G = (V, E)$. The known cactus-tree model [2] represents the minimum weight cuts of G in $O(|V|)$ space. It is used in related studies. Several extensions were suggested for near minimum cuts. For arbitrary nonnegative weights,

Benczur [1] gave an $O(|V|^2)$-space geometric representation of the cuts of weight less than $\frac{6}{5}\lambda$. For integral weights, Dinitz and Nutov [3] suggested two $O(|V|)$-space structures that represent minimum and minimum+1 cuts: one for λ odd, and the other for λ even.

Let us call a cut of a weighted graph *subminimum* if its weight is the second minimum. In [4], we suggested an $O(|V|)$-space representation for minimum and subminimum cuts, provided the weight of a subminimum cut is less than $\frac{4}{3}\lambda$. This structure generalizes and unifies the ones for minimum and minimum+1 cuts of an integrally weighted graph. The construction is based on a new characterization of near-minimum edge cut families that are modeled by a special type of a cactus-tree.

References

[1] A. A. Benczur. The structure of near-minimum edge cuts. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC'95)*, pages 92–102, 1995.

[2] E. A. Dinic, A. V. Karzanov, and M. V. Lomonosov. On the structure of the system of minimum edge cuts in a graph. In *Studies in Discrete Optimization*, pages 290–306. Nauka, Moscow, 1976. In Russian.

[3] Y. Dinitz and Z. Nutov. A 2-level cactus-tree tree model for the minimum and minimum+1 edge cuts in a graph and its incremental maintenance. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC'95)*, pages 509–518, 1995.

- [4] Z. Nutov. On subminimum edge cuts of a weighted graph and cycle uniform cactus-tree models. Manuscript.

Average-Case Complexity of Shortest-Paths Problems

Investigators: Kurt Mehlhorn, Volker Priebe, and C. R. Subramanian

We studied the average-case complexity of shortest-paths problems on different classes of random n-vertex graphs, including graphs that are generated according to the usual $\mathcal{G}(n,p)$ model (with unit and non-negative integer edge weights) and complete graphs with arbitrary real edge weights.

In the $\mathcal{G}(n,p)$ model, each possible edge of an n-vertex graph is chosen to exist with probability p, independent of all the other edges. In [7], we showed that for any p, the all-pairs shortest-paths problem can be solved in $O(n^2 \log n)$ time with high probability (that is, with failure probability $O(n^{-c})$ for an arbitrary constant c). The algorithm uses as a subroutine an $O(n^2 \log n)$-time algorithm for finding witnesses when two Boolean matrices (one or both of which are random) are multiplied. We obtained the following structural result about random graphs in the $\mathcal{G}(n,p)$ model: For $p \geq 18(\log n)/n$, between every pair of vertices, there exists with high probability a path consisting of exactly $\lceil (\log n)/(\log \log n) \rceil$ edges. This bounds the diameter. This proof is based on Janson inequalities [3].

In [8], we extended the work of [7] to the case of integer weights. We assume that the input graph is generated according to the $\mathcal{G}(n,p)$ model and that each existing edge is assigned a weight that is drawn uniformly from $\{0, \ldots, n-1\}$. For such a graph, we showed how to solve both the all-pairs shortest-paths problem and the all-pairs bottleneck-paths problem in $O(n^2 \log n)$ time with high probability. In the *bottleneck-path problem*, the weight of a path is the maximum weight of any edge in it. In the process of analyzing these algorithms, we derive a structural result on the existence of paths with a specified number of edges and bounded weight in such graphs. The analysis requires a new probabilistic tool, namely a weighted analogue (WJI) of the Janson inequalities. The proof of WJI is based on the FKG inequalities. WJI is useful in analyzing "mostly independent" random

events. Even though one can apply the classical second-moment method, the failure probabilities guaranteed by this method are not as low as those guaranteed by WJI.

Work is being carried out on efficiently finding shortest paths in *semi-random graphs*. Semi-random graphs are a generalization of random graphs and this notion was introduced as a way of striking a balance between random graphs and worst-case adversaries.

We also continued our investigation of the average-case complexity of shortest-paths problems on graphs with arbitrary real edge weights. In [1], we studied the *vertex-potential model*, which is a family of probability distributions on complete directed graphs with arbitrary real edge weights but without negative cycles. In the vertex-potential model, there is a (non-negative) random variable $r_{i,j}$ for each edge (i,j), $i,j \in [n]$, and a (real) potential π_i for each vertex $i \in [n]$. Note that we allow the potentials to be arbitrarily chosen; this is a considerable generalization of the model we studied earlier. The weight $c_{i,j}$ of each edge (i,j) is defined by $c_{i,j} = r_{i,j} - \pi_i + \pi_j$. Of course, only the $c_{i,j}$'s are revealed to our algorithms and the $r_{i,j}$'s and π_i's are hidden parameters of the model.

We described two algorithms that are tailored to inputs generated according to the vertex-potential model. They solve the single-source shortest-paths problem in $O(n^2)$ expected time and the all-pairs shortest-paths problem in $O(n^2 \log n)$ expected time, respectively. In both cases our algorithms are reliable, that is, finish their computations within the respective time bounds with high probability.

Another, quite different, probability distribution on graphs with arbitrary real edge weights has been proposed by Kolliopoulos and Stein in [4]. They give a reliable algorithm for the single-source shortest-paths problem with expected running time $O(n^2 \log n)$. Their probabilistic analysis is based on ours in [5]. Our results on lower bounds (in the average case) for the single-source shortest-paths problem from [5] have been extended by Galli [2].

A complete overview of our results is given in [6] (to be submitted shortly).

References

- [1] C. Cooper, A. Frieze, K. Mehlhorn, and V. Priebe. Average-case complexity of shortest-paths problems in the vertex-potential model. *Random Structures Algorithms*. To appear.

[2] N. Galli. *Search processes and their average case analysis*. PhD thesis, Eidgenössische Technische Hochschule, Zürich, Switzerland, 1998.

[3] S. Janson. Poisson approximations for large deviations. *Random Structures Algorithms*, 1(2):221–229, 1990.

[4] S. G. Kolliopoulos and C. Stein. Finding real-valued single-source shortest paths in $o(n^3)$ expected time. *Journal of Algorithms*, 28:125–141, 1998.

[5] K. Mehlhorn and V. Priebe. On the all-pairs shortest-path algorithm of Moffat and Takaoka. *Random Structures Algorithms*, 10:205–220, 1997.

- [6] V. Priebe. *Probabilistic analysis of combinatorial algorithms*. PhD thesis, Universität des Saarlandes, Saarbrücken, 1999. In preparation.

- [7] C. P. Schnorr and C. R. Subramanian. Almost optimal (on the average) combinatorial algorithms for Boolean matrix product witnesses, computing the diameter. In M. Luby, J. Rolim, and M. Serna, editors, *Randomization and Approximation Techniques in Computer Science*, volume 1518 of *Lecture Notes in Compututer Science*, pages 218–231. Springer-Verlag, Berlin, 1998.

- [8] C. R. Subramanian. A generalization of Janson inequalities and its application to finding shortest paths. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 795–804, 1999.

Resource Constrained Shortest Paths

Investigators: Kurt Mehlhorn and Mark Ziegelmann

The resource constrained shortest path problem (RCSP) is the problem of finding the shortest path between two nodes in a graph whenever the traversal of an edge consumes certain resources and the resources consumed along the path must lie within given limits. The problem may be illustrated as that of a traveler with a budget (a vector of resources) who has to reach a given destination as quickly as possible within the constraints imposed by its budget. RCSP has numerous applications in operations research and mission planning. Regarding the complexity, it can be shown that RCSP is NP-hard; however, there is an FPTAS for the problem [4].

Early work dealing with the exact solution of RCSP was done by Joksch [5] who gave a dynamic programming formulation. Other possibilities are to solve the corresponding ILP formulation or to simply adopt a path ranking procedure [2]. However, for large problem instances these approaches are often not efficient. Handler and Zang [3] presented an algorithm for the single resource case based upon Lagrangean relaxation and subsequent path ranking to find the original optimum. Beasley and Christofides [1] solved the multiple resources problem with a subgradient procedure and a subsequent tree search procedure also using problem reductions.

In [6], we considered RCSP for k resources and developed a method that, under certain conditions, efficiently solves a relaxed LP formulation. We associate each path with a point in $k+1$-dimensional space and try to find the facet of the lower hull that intersects the feasible region at the bottommost point. This matches the result in [3] for the single resource case and often results in better upper and lower bounds than [1] for multiple resources and thus provides a basis to solve multiple resource RCSP more efficiently.

We are currently implementing the different approaches to get an experimental comparison of the methods.

References

[1] J. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394, 1989.

[2] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1999.

[3] G. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10:293–310, 1980.

[4] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

[5] H. Joksch. The shortest route problem with constraints. *Journal of Mathematical Analysis and Application*, 14:191–197, 1966.

- [6] K. Mehlhorn and M. Ziegelmann. Resource constrained shortest paths. Manuscript., 1999.

Tree Decomposition

Investigator: Torben Hagerup

The previous progress report described work of ours concerning the parallel construction of tree decompositions of graphs of bounded treewidth and the solution of dynamic problems on such graphs. Tree decompositions will not be defined here; for the following discussion it suffices to know that a tree decomposition of a graph is a tree characterized by two parameters, its *width* and its *diameter*, with the treewidth of a graph G being the smallest width of any tree decomposition

of G. In a traditional sequential setting, having a tree decomposition of small width is crucial, since the complexity of almost anything that one may want to do with the tree decomposition is exponential in the width. In contrast, the diameter of the tree decomposition is of no importance. In parallel and dynamic settings, the width of a tree decomposition is as important as in the sequential case, but the diameter of the decomposition also acquires importance, because the execution time (in a parallel setting) or the query and update times (in a dynamic setting) are proportional to the diameter.

It was known before the research reported here that every n-vertex graph of treewidth k has a tree decomposition of diameter $O(\log n)$ and width at most $3k+2$ [1]. By allowing a width slightly larger than the minimum width, we can thus reduce the diameter to a very low value. However, enlarging the width ever so slightly is highly undesirable because of the exponential dependence of running time on width mentioned above. It is therefore natural to ask whether it is really necessary to go from width k to width $3k+2$ in order to ensure a logarithmic diameter. In [2] we answered this question completely. The answer turns out to be, "Almost, but not quite". More precisely, we show that a logarithmic diameter can be preserved while the width is reduced to $3k-1$, whereas it is impossible in general to achieve a logarithmic diameter and a width of $3k-2$ simultaneously. More generally, we investigate the complete tradeoff between width and diameter of tree decompositions. For all integers n, k and K with $1 \leq k \leq K$, let $D(n,k,K)$ be the maximum, over all n-vertex graphs G of treewidth k, of the minimum diameter of a tree decomposition of G of width at most K. We determine $D(n,k,K)$, up to a constant factor, for all combinations of n, k and K. When K is bounded by a constant (the case of greatest practical relevance), $D(n,k,K)$ is $\Theta(n)$ for $K \leq 2k-1$, $\Theta(\sqrt{n})$ for $2k \leq K \leq 3k-2$, and $\Theta(\log n)$ for $K \geq 3k-1$.

References

[1] H. L. Bodlaender. NC-algorithms for graphs with small treewidth. In *Proceedings of the 14th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'89)*, volume 344 of *Lecture Notes in Computer Science*, pages 1–10. Springer, Berlin, 1989.

- [2] H. L. Bodlaender and T. Hagerup. Tree decompositions of small diameter. In *Proceedings of the 23th International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 702–712. Springer, Berlin, 1998.

Average-Case Analysis of NP-hard Graph Problems

Investigator: C. R. Subramanian

Several graph problems (like k-coloring, or finding a Hamilton cycle) are NP-hard but are solvable in polynomial time over suitable random instances. Even though random graphs tend to possess structural properties not necessarily present in a general graph, an algorithmic study of random graphs could lead to insights into how difficult the problem is. While the best known approximation algorithms can only give a guarantee of $\widetilde{O}(n^{0.25})$ colors on 3-colorable graphs with n vertices, it is long known that random k-colorable graphs can be k-colored almost surely in polynomial time. For some distributions, this can also be done in polynomial average time (*p.av.t.*) [3, 6].

Consider random graphs drawn by choosing each allowed edge independently with probability p after initially partitioning the vertex set into k color classes of "roughly equal" (*i.e.* $\Omega(n)$) sizes. We obtained a simple BFS-tree based approach [7] to separate the largest or smallest color class. Repeating this procedure $k-1$ times, one obtains a k-coloring of G with high probability, if $p \geq n^{-1+\epsilon}$, $\epsilon \geq X/\sqrt{\log n}$ for some sufficiently large constant X. Our algorithms have very small failure probabilities. They are also much simpler than previous algorithms based on spectral techniques

[1] or semi-definite programming [4]. An additional strength is that our algorithms can be further modified so as to have much lower failure probabilities at the cost of running time. This helped us in obtaining *p.av.t.* algorithms for the same range of p improving the previous results on *p.av.t.*coloring [6] where ϵ is required to be above $1/4$.

The BFS approach outlined before seems applicable to other partitioning problems also. Using this approach, we were able to show how to obtain a minimum bisection in random graphs [8]. Consider a random graph on $2n$ vertices generated as follows. Partition the vertices into A and B of size n each and include every edge joining A and B with probability q and every other edge with probability $p \geq q$. For q sufficiently smaller than p, $A : B$ will be the unique optimal bisection almost surely. Given such a graph, we prove that a simple algorithm based on growing BFS trees outputs $A : B$ almost surely, provided p is in the same range mentioned before. In simplicity, it is comparable to the Metropolis algorithm of [5]; still it works for a class of distributions (determined by p and $p - q$) almost comparable to those required by the more sophisticated algorithms based on ellipsoid method [2] and semi-definite programming [4]. In particular, for $p = \tilde{\Omega}(n^{-0.5})$, the algorithm succeeds even if $p - q$ becomes as small as $n^{-0.5+\delta}$ for any small constant δ. As $p - q$ approaches $n^{-0.5}$, $A : B$ ceases to be the unique optimal bisection.

References

[1] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC'94)*, pages 346–355, 1994.

[2] R. B. Boppana. Eigenvalues and graph bisection: an average-case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (FOCS'87)*, pages 280–285, 1987.

[3] M. E. Dyer and A. M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10:451–489, 1989.

[4] U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98)*, pages 674–683, 1998.

[5] M. Jerrum and G. B. Sorkin. Simulated annealing for graph bisection. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS'93)*, pages 94–103, 1993.

[6] C. R. Subramanian. Minimum coloring random and semi-random graphs in polynomial expected time. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 463–472, 1995. Also a journal submission.

- [7] C. R. Subramanian. Algorithms for coloring sparse random graphs. Manuscript, 1999.
- [8] C. R. Subramanian. Graph bisection on random graphs. Manuscript, 1999.

Robustness Analysis

Investigator: Roberto Solis-Oba

A fundamental problem in the study of dynamic systems is that of measuring how sensitive a problem is to perturbations in its input. In some situations it is necessary to determine the maximum effect that bounded changes in the whole input of a problem can have over the value of its solution, so that sensitivity analysis does not suffice. In [1, 2], we considered the important class of matroid optimization problems, and presented efficient algorithms to compute the effect that this type of change has over their solutions. The *perturbability function* of a matroid measures the maximum increase in the weight of its minimum weight bases that can be produced by increases of a given total cost on the weights of its elements.

Matroid theory provides an elegant structure that captures the essence of a large and important class of problems [3]. There are matroid optimization problems in computational biology, graph theory, and electrical networks [4] for which only estimates of the input values are available, or for which changes in the input values are expected. The perturbability functions for these problems can be used to assess the quality of their solutions. In [2], we presented a general algorithm for computing the perturbability function for any matroid. Our algorithm computes the perturbability function of a weighted matroid in $O(m^5n^2 + m^4n^4\tau)$ time, where m is the number of elements in the matroid, n is its rank, and τ is the time needed to test independence for a set of at most n elements. As we show, the perturbability function is piecewise linear and it has at most mn breakpoints. Our algorithm can compute all the breakpoints of the function within the time bound stated above. For the case of transversal matroids, we gave an algorithm that computes the perturbability function in $O(mn(m+n^2)|E|\log(m^2/|E|+2))$ time, where E is the set of edges in the bipartite graph that defines the transversal matroid.

References

[1] G. N. Frederickson and R. Solis-Oba. Efficient algorithms for robustness in matroid optimization. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 659–668, 1997.

- [2] G. N. Frederickson and R. Solis-Oba. Algorithms for measuring perturbability in matroid optimization. *Combinatorica*, 18:503–518, 1999.

[3] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, 1976.

[4] A. Recski. *Matroid Theory and its Applications.* Springer, Berlin, 1989.

2.3 Implementing Algorithms

Fun with implementing algorithms

Investigator: Rudolf Fleischer

Inventing new algorithms only makes sense if they are also put to use, *i.e.*, if they are implemented and used to solve some real problems. For many years, theoretical computer scientists have not bothered to take this side of the trade seriously. This has only recently changed to some extent [2]. In [1], we argued that implementing algorithms is more than fun: it sharpens the view on the fine points of an algorithm or the problem itself; it can help to avoid publishing embarrassingly erroneous results; and well animated algorithms can be used in teaching algorithms to supplement the otherwise dry theoretical explanations.

References

- [1] R. Fleischer. FUN with implementing algorithms. In *Proceedings of* FUN with Algorithms*, Elba, Italy.* Carleton Scientific, 1998.

- [2] K. Mehlhorn and S. Näher. *LEDA – A Platform of Combinatorial and Geometric Computing.* Cambridge University Press, Cambridge, England, 1999. To appear. A preview is possible at `http://www.mpi-sb.mpg.de/~mehlhorn/LEDAbook.html`.

An Experimental Study of Dynamic Algorithms for Directed Graphs

Investigator: Christos Zaroliagis

Dynamic graph algorithms have been an active and blossoming research field over the last years. Many important theoretical results have been obtained for both fully and partially dynamic maintenance of several properties on undirected graphs. Recently, an equally important effort has started to implement these techniques and show their practical merits. For directed graphs (digraphs), the development of fully dynamic algorithms turned out to be a very hard problem and most of the research so far has concentrated on the design of partially dynamic algorithms. However, despite the number of interesting theoretical results achieved, very little has been done so far w.r.t. implementations even for the most fundamental problems.

In [3], we have made a step forward in bridging the gap between theoretical results for directed graphs and their implementation by conducting an experimental study of several dynamic algorithms for transitive closure on digraphs (constituting the bulk of our work) as well as for depth first search (DFS) and topological sorting on directed acyclic graphs (DAGs). For transitive closure, we have implemented the algorithms proposed in [1, 4, 5, 6, 8], plus several variants of them, and several simple-minded algorithms that were easy to implement and likely to be fast in practice. We also developed a new algorithm, which is a variant of Italiano's algorithms [5, 6] and whose decremental part applies to any digraph, not only to DAGs. For DFS and topological sorting we have implemented the incremental algorithms for DAGs in [2, 7], and developed and implemented a new simple decremental algorithm for maintaining a DFS tree in a DAG. Our experiments have been performed on several kinds of random inputs, non-random inputs that are worst-case inputs for the dynamic algorithms, and a real world graph (fragment of the Internet network).

For transitive closure and in the case of random inputs, a fine-tuned version of Italiano's algorithms [5, 6] as well as our new variant of these algorithms were almost always the fastest in the incremental case as well as in the decremental case for DAGs. In the decremental case for general digraphs, the simple-minded algorithms were always significantly faster than the decremental algorithm of [4] or our new variant of Italiano's algorithms. A similar behavior was observed in the fully dynamic case for general digraphs; a (perhaps not) surprising fact was that the theoretically fastest fully dynamic algorithm of [4] was the slowest in practice, even for a very small sequence of operations (*e.g.*, 10). In the fully-dynamic case for DAGs, again the fine-tuned version or our new variant of Italiano's algorithms was the fastest when the initial graph was not sparse; in the sparse case, the simple-minded algorithms became competitive. Our experiments with the internet graph gave similar conclusions to those obtained with the random inputs. In the case of non-random inputs, the simple-minded algorithms were significantly faster than any of the dynamic algorithms. In the case of topological sorting and DFS,, our experiments showed that the dynamic algorithms considered were always faster than recomputing from scratch, especially for sparse graphs.

References

[1] S. Cicerone, D. Frigioni, U. Nanni, and F. Pugliese. A uniform approach to semi dynamic problems in digraphs. *Theoretical Computer Science*. To appear.

[2] P. Franciosa, G. Gambosi, and U. Nanni. The incremental maintenance of a depth-first-search tree in directed acyclic graphs. *Information Processing Letters*, 61(2):113–120, 1997.

- [3] D. Frigioni, T. Miller, U. Nanni, G. Pasqualone, G. Schaefer, and C. Zaroliagis. An experimental study of dynamic algorithms for directed graphs. In *Proceedings of the 6th European Symposium on Algorithms (ESA'98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 368–380. Springer, 1998.

[4] M. R. Henzinger and V. King. Fully dynamic biconnectivity and transitive closure. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS'95)*, pages 664–672, 1995.

[5] G. F. Italiano. Amortized efficiency of a path retrieval data structure. *Theoretical Computer Science*, 48:273–281, 1986.

[6] G. F. Italiano. Finding paths and deleting edges in directed acyclic graphs. *Information Processing Letters*, 28:5–11, 1988.

[7] A. Marchetti-Spaccamela, U. Nanni, and H. Rohnert. Maintaining a topological ordering under edge insertions. *Information Processing Letters*, 59(1):53–58, 1996.

[8] D. M. Yellin. Speeding up dynamic transitive closure for bounded degree graphs. *Acta Informatica*, 30:369–384, 1993.

Animated Algorithms

Investigator: Rudolf Fleischer

We continued our effort to build a library of animated online algorithms, the OnVis system (*On*line *Vis*ualization system). We completely redesigned the central server to make it more user-friendly than the old prototype OnVis 1.0. A first version of OnVis 2.0 is now available [1]. We also implemented more online algorithms like, for example, algorithms for list update, bin packing, graph exploration, localization in trees, etc.

Recently, we also started another project where we would like to animate all elementary data structures and algorithms of the LEDA library. This can serve as an animated LEDA manual over the web, and will also quite likely become a helpful teaching tool to supplement the otherwise quite dry theoretical descriptions of algorithms.

References

- [1] R. Fleischer and P. Hyska. OnVis 2.0: A system for visualisation and animation of online algorithms. Technical report, Max-Planck-Institut für Informatik, Saarbrücken, Germany. In preparation.

3 Approximation and Online Algorithms

The activity of our group in the area of approximation and online algorithms has considerably increased during the last two years. Our research in this field focuses mainly on theoretical issues; however, some of the results have already been coupled with experimental work.

For a wide variety of important optimization problems, finding an optimal solution is computationally hard. In fact, for many of these problems, there is strong (mathematical) evidence that shows that – except for instances of small size – there does not exist an efficient algorithm to find optimal solutions. A common way of dealing with such problems is to design efficient algorithms that compute provably "good" approximate solutions, *i.e.*, feasible solutions that approximate the optimum within a small relative accuracy. This trading of optimality for tractability is the main paradigm of approximation algorithms.

The members of our group have approached a great variety of optimization problems from the point of view of approximation. We considered network flow problems and provided new combinatorial polynomial-time algorithms for computing approximate multicommodity and unsplittable flows. We also studied integrated prefetching and caching, and gave a polynomial-time D-approximation algorithm for the stall time minimization problem with D parallel disks. For the general caching problem, where the requested pages can be of different sizes and costs, we proved that constant approximation can be achieved in polynomial time when it is allowed to use a limited amount of additional memory. For dynamic storage allocation, we developed a fast 3-approximation algorithm. Different optimization problems on graphs were also examined by members of our group. In particular, constant approximation algorithms were presented for finding spanning trees in graphs with a maximum number of leaves and computing the achromatic number of trees. There was extensive work by group members on scheduling. This includes linear-time approximation schemes for various preemptive and non-preemptive makespan minimization problems with any fixed number of machines including scheduling on unrelated parallel machines and job shop, as well as malleable and non-malleable multiprocessor task scheduling. We have also obtained interesting results on binpacking, wavelength assignment in optical networks, and node-connectivity problems.

Online computation is decision-making with incomplete information. Typically, an online algorithm receives a sequence of *requests* and must react immediately to each of these requests without knowing future requests. An online algorithm A is called *c-competitive* if, for every request sequence, the solution computed by A is at most a factor of c away from an optimal solution for that sequence. During the last two years we have investigated a large variety of online problems.

A fundamental problem that we have examined is *delay* in online algorithms. In a delay model, at any time-step, an online algorithm does not have up-to-date information for that time-step and/or a decision taken at that time-step does not have an immediate effect. We have studied several basic online problems with respect to delay and gave tight or nearly tight analyses. We have introduced the *Bahncard* problem which is a generalization of the well-known ski rental problem. The Bahncard is a railway pass of the German railway company that entitles the holder to a 50% price reduction on tickets. We have devised optimal and nearly optimal deterministic and randomized strategies for buying a Bahncard. Another line of research concentrates on online scheduling. We consider a basic problem where jobs have to be scheduled on m identical machines so as to minimize the makespan. For small machine numbers m ($m = 2, 3$), we have developed barely random algorithms that use only a constant number of random bits regardless of the length of the input sequence. Additionally, we presented improved bounds for the problem variant where jobs may be rejected. Other online problems that we have addressed include robot exploration, page replication, bin packing and call admission.

3.1 Approximation Algorithms

Scheduling Unrelated Parallel Machines

Investigators: Klaus Jansen and Lorant Porkolab

A fundamental problem in scheduling theory is that of scheduling n independent jobs on m unrelated parallel machines. Each job has to be processed by exactly one machine, and processing job j on machine i requires p_{ij} time units. The objective is to minimize the makespan, *i.e.* the maximum job completion time.

This problem is known to be NP-hard even when there are only 2 machines. Lenstra, Shmoys and Tardos [3] gave a polynomial-time 2-approximation algorithm for the problem, and this is still the currently best approximation ratio achieved in polynomial time. They also proved that for any $\epsilon < 1/2$, no polynomial-time $(1+\epsilon)$-approximation algorithm exists, unless P=NP. When the number of machines m is fixed, the problem becomes easier. Horowitz and Sahni [1] proved that for any $\epsilon > 0$, an $(1+\epsilon)$-approximate solution can be computed in $O(nm(nm/\epsilon)^{m-1})$ time. Lenstra *et al.*g [3] also gave an approximation scheme for the problem with a running time which is bounded by a product of $(n+1)^{m/\epsilon}$ and a polynomial of the input size.

In [2] we present a new approximation scheme for the problem whose running time is $n(m/\epsilon)^{O(m)}$. If there are only a constant number of machines, it gives a fully polynomial-time approximation scheme which computes for any fixed $\epsilon > 0$ an $(1+\epsilon)$-approximate solution in $O(n)$ time. This linear complexity bound is a substantial improvement in terms of n compared to the above mentioned results. For the preemptive version of the general problem we propose a fully polynomial approximation scheme whose running time depends also only linearly on n.

Recently a variant of the above problem was considered, where processing job j on machine i incurs a cost of c_{ij}. In this problem the objective is to find a schedule of bounded makespan and cost. Shmoys and Tardos [4] designed a polynomial-time algorithm that, given values C and T, finds a schedule with cost at most C and makespan at most $2T$, if a schedule of cost C and makespan at most T exists. We designed an approximation scheme [2] that, given values T and C, computes in $n(m/\epsilon)^{O(m)}$ time for any $\epsilon > 0$ a schedule of length at most $(1+\epsilon)T$ and cost at most $(1+\epsilon)C$, if there exists a schedule of makespan T and cost C. When m is constant, this algorithm is a fully polynomial-time approximation scheme that runs in $O(n)$ time for any fixed $\epsilon > 0$.

References

[1] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, pages 317–327, 1976.

- [2] K. Jansen and L. Porkolab. Improved approximation schemes for scheduling unrelated parallel machines. In *Proceedings 31st Annual ACM Symposium on Theory of Computing (STOC'99)*, 1999. To appear.

[3] J.K. Lenstra, D.B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.

[4] D.B. Shmoys and E. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993.

Job Shop Scheduling

Investigators: Klaus Jansen and Roberto Solis-Oba

In the job shop scheduling problem, there is a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n jobs that must be processed on a group $M = \{1, \ldots, m\}$ of m machines. Each job J_j consists of a sequence of operations $O_{1j}, O_{2j}, \ldots, O_{\mu j}$, where Operation O_{ij} must be processed without interruption on machine $m_{ij} \in \{1, \ldots, m\}$ during p_{ij} time units. The operations $O_{1j}, O_{2j}, \ldots, O_{\mu j}$ must be processed one after another in the given order and each machine can process at most one operation at a time.

The job shop scheduling problem is considered to be one of the most difficult problems in combinatorial optimization. Even very constrained versions of the problem are strongly NP-hard. Two other widely studied shop scheduling problems are the flow shop and the open shop problems. Williamson *et al.*g [6] proved that for any $\rho < 5/4$, the existence of a ρ-approximation algorithm for any of the above shop scheduling problems would imply that P=NP.

For the case that the number m of machines is fixed, Hall [1] has developed a polynomial time approximation scheme for flow shops, while Sevastianov and Woeginger [5] designed an $O(n \log n)$ time approximation scheme for open shops. For the case of job shops with fixed m and μ, we have designed a linear time approximation scheme [3, 2].

The idea behind our algorithm is to divide the set of jobs $\mathcal{J}$ into two groups $\mathcal{L}$ and $\mathcal{S}$ formed by jobs with "large" and "small" total processing time, respectively. We fix a relative ordering for the long jobs and find a schedule for the small jobs using a linear program. Then we round the solution for the linear program so that only a constant number of small jobs are preempted. These jobs are scheduled sequentially at the end of the solution. The linear program induces a partition on the set of small jobs, and the rounded solution might be infeasible in each group of this partition. We use Sevastianov's [4] algorithm independently on each group to find a feasible schedule for the whole set of jobs.

We show that the approximation scheme can be generalized also to the preemptive version of the job shop scheduling problem and to the job shop problem with release and delivery times. We can also handle more general problems, like the flexible job shop problem, in which there is a group of machines that can process any job, and the dag job shop problem, in which only a partial order is specified for the execution order of the operations of a job.

References

[1] L. A. Hall. Approximability of flow shop scheduling. *Mathematical Programming*, 82:175–190, 1998.

- [2] K. Jansen, R. Solis-Oba, and M.I. Sviridenko. A linear time approximation scheme for the job shop scheduling problem. Manuscript, 1999.

- [3] K. Jansen, R. Solis-Oba, and M.I. Sviridenko. Makespan minimization in job shops: a polynomial time approximation scheme. In *Proceedings 31st Annual ACM Symposium on Theory of Computing (STOC'99)*, 1999. To appear.

[4] S. V. Sevastianov. Bounding algorithm for the routing problem with arbitrary paths and alternative servers. *Cybernetics*, 22:773–780, 1986.

[5] S. V. Sevastianov and G. J. Woeginger. Makespan minimization in open shops: A polynomial time approximation scheme. *Mathematical Programming*, 82:191–198, 1998.

[6] D. P. Williamson, L. A. Hall, J. A. Hoogeveen, C. A. J. Hurkens, J. K. Lenstra, S. V. Sevast'janov, and D. B. Shmoys. Short shop schedules. *Operations Research*, 45:288–294, 1997.

Scheduling Malleable Parallel Tasks

Investigators: Klaus Jansen and Lorant Porkolab

The problem of scheduling malleable parallel tasks is defined as follows. Let $\mathcal{T} = \{T_0, \ldots, T_{n-1}\}$ be a set of tasks, let m identical processors be given, and let $M = \{1, \ldots, m\}$. Each task T_j has an associated function $t_j : M \to Q^+$ that gives the execution time $t_j(\ell)$ of task T_j in terms of the number of processors ℓ that are assigned to T_j. If β_j processors are alloted to task T_j, all these processors are required to execute task T_j in union and without preemption. A feasible non-preemptive schedule consists of a processor allotment and a starting time $\tau_j \geq 0$ for each task T_j such that at each time τ, the number of active processors does not exceed the total number of processors. The objective is to find a feasible non-preemptive schedule that minimizes the overall makespan.

This problem has been studied in several recent papers, see *e.g.* [1, 3]. The problem of scheduling non-malleable parallel tasks is a restriction of the above problem in which the processor allotments are known a priori. Closely related problems are rectangle packing and resource constrained scheduling. The problems of scheduling malleable and non-malleable parallel tasks are strongly NP-hard even when the number of processors is constant [1], but their optimum can be approximated within a factor of 2 [3].

In [2] we studied the problem under the assumption that there are only a constant number of processors. We designed linear time approximation schemes for both malleable and non-malleable parallel task scheduling. The algorithm first computes $d_j = \min_{\ell=1,\ldots,m} t_j(\ell)$ for each task T_j and selects a constant number $k = k(m, \epsilon)$ of tasks $T_{j_1}, \ldots, T_{j_k}$ with the largest d_j values. Next, it constructs all relative schedules for the set $\mathcal{L} = \{T_{j_1}, \ldots, T_{j_k}\}$. For each relative schedule, we consider an integer program for scheduling all tasks in $\mathcal{T}$ such that the relative schedule of $\mathcal{L}$ is respected. The linear programming relaxation of this integer program can be decomposed into two parts: a fractional packing problem and a linear program with a constant number of variables and constraints. By using this decomposition and an approximation scheme for fractional packing problems, the algorithm solves the linear programming relaxation approximately. We show that the makespan of the schedule produced by this procedure is at most $(1 + \epsilon)$ times the optimum.

References

[1] J. Du and J. Leung. Complexity of scheduling parallel task systems. *SIAM Journal on Discrete Mathematics*, 2:473–487, 1989.

- [2] K. Jansen and L. Porkolab. Linear-time approximation schemes for scheduling malleable parallel tasks. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 490–498, 1999.

[3] W. Ludwig and P. Tiwari. Scheduling malleable and nonmalleable parallel tasks. In *Proceedings 5th ACM-SIAM Symposium on Discrete Algorithms (SODA'94)*, pages 167–176, 1994.

Mutual Exclusion Scheduling

Investigator: Klaus Jansen

The following problem arises in scheduling theory: there are n jobs that must be completed on m processors in minimum time t. A processor can execute only one job at a time, and each job requires one time unit for completion. The scheduling is complicated by additional resource requirements(*e.g.* I-O devices, communication links). A job can only be scheduled on a processor in a given time unit after it has an exclusive lock on all required resources. One application arises in

load balancing the parallel solution of partial differential equations (pde's) by domain decomposition [1]. The domain for the pde's is decomposed into regions where each region corresponds to a subcomputation. The subcomputations are scheduled on m processors so that subcomputations corresponding to regions that touch at even one point are not performed simultaneously.

These scheduling problems can be solved by creating an undirected graph $G = (V, E)$ with a vertex for each of the n jobs, and an edge between every pair of conflicting jobs. In each time step, we can execute any subset $U \subset V$ of jobs for which $|U| \leq m$ and U is an independent set in G. A minimum length schedule corresponds to a partition of V into a minimum number t of such independent sets. Baker and Coffman called this graph-theoretical problem Mutual Exclusion Scheduling.

In [2] we proved the following result: For each constant $m \geq 6$, the Mutual Exclusion problem is NP-complete for permutation and also for comparability graphs. Finding the complexity for smaller constants ($m = 3, 4, 5$) could be a step to the solution of the famous open m-machine scheduling problem with unit times.

References

[1] B. S. Baker and E. G. Coffman. Mutual exclusion scheduling. *Theoretical Computer Science*, 162:225 – 243, 1996.

- [2] K. Jansen. The mutual exclusion scheduling problem for permutation and comparability graphs. In *Proceedings Symposium on Theoretical Aspects of Computer Science (STACS'98)*, pages 287 – 297. LNCS 1373, Springer Verlag, 1998.

Network Flow Problems

Investigators: Naveen Garg and Jochen Könemann

In [2] we consider the problem of designing fast, combinatorial approximation algorithms for multicommodity flows and other fractional packing problems. We provide a new approach to these problems, which yields faster and much simpler algorithms. In particular, we provide the first polynomial-time, combinatorial approximation algorithm for the fractional packing problem whose running time is strongly polynomial. Our approach also allows us to substitute shortest path computations for min-cost flow computations in computing maximum concurrent flow and min-cost multicommodity flow; this yields much faster algorithms when the number of commodities is large.

In [1] we studied another network flow problem. Let $G = (V, E)$ be a capacitated directed graph with a source s and k terminals t_i with demands d_i, $1 \leq i \leq k$. We would like to concurrently route every demand on a single path from s to the corresponding terminal without violating the capacities. There are several interesting and important variations of this unsplittable flow problem.

If the necessary cut condition is satisfied, we show how to compute an unsplittable flow satisfying the demands such that the total flow through any edge exceeds its capacity by at most the maximum demand. For graphs in which all capacities are at least the maximum demand, we therefore obtain an unsplittable flow with congestion at most 2, and this result is the best possible. Furthermore, we show that all demands can be routed unsplittably in 5 rounds, *i.e.*, all demands can be collectively satisfied by the union of 5 unsplittable flows. Finally, we show that 22.6% of the total demand can be satisfied unsplittably.

These results are extended to the case when the cut condition is not necessarily satisfied. We derive a 2-approximation algorithm for congestion, a 5-approximation algorithm for the number of rounds, and a $4.43 = 1/0.226$-approximation algorithm for the maximum routable demand.

References

- [1] Y. Dinitz, N. Garg, and M.X. Goemans. On the single-source unsplittable flow problem. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 290–299, 1998.
- [2] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 300–309, 1998.

Integrated Prefetching and Caching

Investigators: Susanne Albers, Naveen Garg, and Stefano Leonardi

Prefetching and caching are powerful tools for increasing the performance of file and database systems. Both tools have separately been the subjects of extensive theoretical and experimental studies. However, only recently have researchers started looking at these techniques in an integrated manner. In a seminal work Cao *et al.g* [2] introduced a model that allows an algorithmic study of the problem. We are given a request sequence σ and a cache of size k. Each request in σ specifies a memory block stored on disk. We emphasize that we study the offline problem in which the entire request sequence is given in advance. Serving a request takes one time-unit. Fetching a block not in cache takes F time-units. If we initiate a prefetch to the block some i requests before the actual reference, then the processor has to stall for $F - i$ time-units. When loading a block into cache we also have to evict a block from cache. The goal is to minimize the total processor stall time incurred in serving the request sequence.

For single disk systems, Cao *et al.g* [2] gave a $\min\{2, 1 + \frac{F}{k}\}$-approximation algorithm. This algorithm approximates the *elapsed time*, which is the sum of the processor stall time and the length of the request sequence. For systems with D parallel disks, Kimbrel and Karlin [3] gave a $(1 + D\frac{F}{k})$-approximation for the elapsed time.

In [1] we present a new approach to the problem of minimizing stall time in single and parallel disk systems. We formulate the problems as integer programs and solve linear relaxations of these programs. We first prove that for single disk systems, an *optimum* prefetching and caching schedule can be comuputed in polynomial time. In particular, we show that any optimum fractional solution of our linear program can be written as a convex combination of (polynomially many) integral solutions. This is equivalent to saying that there is an optimum solution to the linear program that is integral. For parallel disk systems, we consider the problem of minimizing the total stall time instead of the total elapsed time. While minimizing these two measures is equivalent, approximating total stall time is harder than approximating elapsed time, since the length of the sequence is not part of our objective function. We give a D-approximation algorithm for minimal total stall time. The solution constructed uses at most $D - 1$ additional memory locations in cache. This is actually very small — D is typically 4 or 5 — when compared with the size of the cache. Note that for $D = 1$, we obtain our optimum algorithm for the single disk case. Another pleasing feature of our algorithm is that, if a sequence can be served with zero stall time, we obtain a schedule that has no stall either and uses at most $D - 1$ extra memory locations in cache.

References

- [1] S. Albers, N. Garg, and S. Leonardi. Minimizing stall time in single and parallel disk systems. In *Proceedings 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, pages 454–462, 1998.

[2] P. Cao, E.W. Felten, A.R. Karlin, and K. Li. A study of integrated prefetching and caching strategies. In *Proceedings ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'95)*, pages 188–196, 1995.

[3] T. Kimbrel and A.R. Karlin. Near-optimal parallel prefetching and caching. In *Proceedings 37th IEEE Annual Symposium on Foundations of Computer Science (FOCS'96)*, pages 540–549, 1996.

General Caching Problems

Investigator: Susanne Albers

Caching (paging) is a well studied problem in online algorithms, usually studied under the assumption that all pages have a uniform size and a uniform fault cost (*uniform caching*). However, recent applications related to the World Wide Web involve situations in which pages can be of different sizes and costs. This *general caching problem* seems more intricate than the uniform version. In particular, the offline case itself is NP-hard. Only a few results exist for the general caching problem. Irani [2] gave offline approximation algorithms for the Bit and the Fault Models. In the Bit Model, the cost of loading a page into cache is equal to the size of the page. In the Fault Model, the loading cost is 1, independent of the size of the page. The approximation ratios achieved by Irani are $O(\log k)$, where k is the ratio of the cache size to the size of the smallest page ever requested.

In [1] we develop improved offline page replacement policies for the general caching problem. Our first main result is that by using only a small amount of additional memory, say $O(1)$ times the largest page size, we can obtain an $O(1)$-approximation to the general caching problem. Note that the largest page size, which we denote by S, is typically a very small fraction of the total cache size, say 1%. In the Bit Model we achieve a 1-approximation, *i.e.* a solution with optimum loading cost, using an extra space of S. In the Fault Model, we obtain a 2-approximation using $2S$ extra space. We also consider the General Cost Model where the loading cost of a page can be arbitrary. We achieve a 10-approximation using $4S$ extra space. More generally, we can develop trade-offs between the approximation ratios and the extra space used. In the Fault Model, the best approximation ratio we obtain is $1 + \epsilon$, for any $\epsilon > 0$. In the General Model, the best ratio is $4 + \epsilon$. Our second result in [1] is that when no additional memory is allowed, one can obtain an $O(\log(M + C))$-approximation where M and C denote the cache size and the largest page fault cost, respectively. Our results use a new rounding technique for linear programs, which may be of independent interest. We also present a randomized online algorithm for the Bit Model which achieves a competitive ratio of $O(\ln(1 + 1/c))$ while using $M(1 + c)$ memory.

References

- [1] S. Albers, S. Arora, and S. Khanna. Page replacement for general caching problems. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 31–40, 1999.

[2] S. Irani. Page replacement with multi-size pages and applications to web caching. In *Proceedings 29th Annual ACM Symposium on Theory of Computing (STOC'97)*, pages 701–710, 1997.

Algorithms for Compile-Time Memory Optimization

Investigator: Jordan Gergov

Given a program in a structured programming language, the compiler can use control-flow analysis techniques to determine pairs of source-code objects (*e.g.* arrays or structures) such that the objects in each pair cannot "interfere" with each other at run-time and, hence, can share memory. The *Compile-Time Memory Allocation* problem (CMA) is to construct a memory allocation for all objects such that the memory usage is minimized and memory regions of objects that do not "interfere" at run-time are allowed to overlap [1].

In [3], we propose the first polynomial-time algorithm for CMA with a performance guarantee. Our approach is based on a novel algorithm for off-line *Dynamic Storage Allocation* (DSA). DSA can be viewed as a special case of CMA and has a number of independent applications, for instance, in channel routing, logistics, and communication protocols. In geometric terms, the input of DSA consists of n rectangles that are described by n triples of numbers $\{(s_1, r_1, c_1), \ldots, (s_n, r_n, c_n)\}$. Each triple (s_i, r_i, c_i) corresponds to an axis-parallel rectangle with a projection (r_i, c_i) on the x-axis and a projection of length s_i on the y-axis. We are only allowed to slide the rectangles along the y-axis while the x-projections of all rectangles stay fixed as in the input. The objective is to pack all rectangles in a horizontal strip of minimum height. We give a new and simple $O(n \log n)$ time 3-approximation for DSA [3]. This result improves the best previous approximation ratio of 5 [2]. The key idea behind our 3-approximation is an incremental construction of a 2-allocation. The concept of 2-allocation was introduced in [2] in order to show that the lower bound on the performance of on-line coloring-based DSA approximations can be improved.

References

[1] J. Fabri. Automatic storage optimization. *ACM SIGPLAN Notices*, 14(8):83–91, 1979.

[2] J.N. Gergov. Approximation algorithms for dynamic storage allocation. In *Proceedings 4th European Symposium on Algorithms (ESA'96)*, pages 52–61, September 1996.

- [3] J.N. Gergov. Algorithms for compile-time memory optimization. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 907–908, January 1999.

Spanning Trees with Maximum Number of Leaves

Investigator: Roberto Solis-Oba

The problem of finding a spanning tree with the maximum number of leaves has applications in the design of communication networks, circuit layouts, and in distributed systems. Galbiati *et al.*g [1] have proven that the problem is MAXSNP-complete. In [5] we presented a 2-approximation algorithm for the problem, improving on the previous best performance ratio of 3 achieved by an algorithm of Ravi and Lu [4].

This problem is, from the point of view of optimization, equivalent to the problem of finding a minimum connected dominating set. But the problems are very different when considering how well their solutions can be approximated. Khuller and Guha [2] show that the minimum connected dominating set problem cannot be approximated within ratio $(1 - o(1)) \ln n$. However, the solution to the problem of finding a spanning tree with the maximum number of leaves is approximable within a constant of the optimum value [4]. There are several papers that deal with the question of determining the largest value ℓ_k such that every connected graph with minimum degree k has a spanning tree with at least ℓ_k leaves [3].

Ravi and Lu [4] introduced the concept of a *leafy forest* that allowed them to design an efficient 3-approximation algorithm for the problem. We improve on the algorithm by Ravi and Lu by providing a linear time algorithm that finds a spanning tree with at least half of the number of leaves in any spanning tree of a given undirected graph. Our algorithm uses *expansion rules*, similar to those in [3]. However, we assign priorities to the rules and use them to build a forest instead of a tree as in [3].

Informally, expansion rules of low priority increase by a small amount the number of leaves in the forest, while rules of high priority increase this number by a large amount. We show that each rule of low priority adds to the forest at least one vertex that must be internal in any optimal

tree T^*. Moreover, we show that this set of internal vertices is different from the internal vertices required to interconnect the subtrees induced in T^* by the vertices spanned by F. By careful implementation, the algorithm can be made to run in linear time.

We also consider the variant of the problem in which a given set of vertices S must be leaves and a spanning tree T_S with maximum number of leaves subject to this constraint is sought. By using the above algorithm, we reduce this problem to a variant of the set covering problem in which instead of minimizing the size of a cover, we want to maximize the number of sets that do not belong to the cover. We present a simple heuristic for this latter problem, which yields a $(5/2)$-approximation algorithm for finding the spanning tree T_S.

References

[1] G. Galbiati, F. Maffioli, , and A. Morzenti. A short note on the approximability of the maximum leaves spanning tree problem. *Information Processing Letters*, 52:45–49, 1994.

[2] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In *Proceedings 4th Annual European Symposium on Algorithms*, pages 179–193, 1996.

[3] D. J. Kleitman and D. B. West. Spanning trees with many leaves. *SIAM Journal on Discrete Mathematics*, 4:99–106, 1991.

[4] H. Lu and R. Ravi. A near-linear time approximation algorithm for maximum-leaf spanning tree. *Journal of Algorithms*, 29:132–141, 1998.

- [5] R. Solis-Oba. 2-approximation algorithm for finding a spanning tree with maximum number of leaves. In *Proceedings 6th Annual European Symposium on Algorithms (ESA'98)*, pages 441–452. LNCS 1461, Springer Verlag, 1998.

Rooted Spanning Trees with Small Weight and Average Length

Investigator: Roberto Solis-Oba

Let $G = (V, E)$ be an undirected graph with non-negative weights on the edges. Let T be a spanning tree of G with root r. We denote by $w(T)$ the sum of the weights of the edges in T. For a vertex $v \in V$, let T_v be the path in T from r to v. The weight of path T_v is the sum of the weights of its edges. The *total length of* T, denoted as $d(T)$, is the sum of the weights of the paths T_v for all vertices $v \in V$.

In [1] we studied the problem of finding a spanning tree of G that minimizes the function $f_\lambda(T) = \lambda w(T) + (1-\lambda)d(T)$ for a given value of the parameter λ, $0 \leq \lambda \leq 1$. Possible applications for this problem include multicasting routing in trees and VLSI design.

Given a rooted spanning tree T, the *multiplicity* $m(e)$ of an edge $e \in T$ is the number of paths from the root to vertices in the tree that use edge e. The total length of T can be written as $d(T) = \sum_{e \in T} m(e)w(e)$. Hence $f_\lambda(T) = \sum_{e \in T}[\lambda + (1-\lambda)m(e)]w(e)$. This form shows explicitly the contribution of every edge to the total value of the function.

There might be situations in which the cost of using edge e is not proportional to its multiplicity. Suppose for example that it is desired to build a tree in which the failure of any link affects only a small set of vertices. In such a tree the multiplicities of the edges should be kept small. To find this tree we can use an objective function that assigns to each edge a cost that is super-linear in its multiplicity.

In [1] we show that the problem of finding a rooted spanning tree T that minimizes the function $f_\lambda(T) = \lambda w(T) + (1-\lambda)d(T)$ is NP-hard for all values $0 < \lambda < 1$. Note that when $\lambda = 1$, the problem reduces to that of finding a minimum spanning tree, and when $\lambda = 0$, the problem

is equivalent to finding a shortest path tree. Interestingly, the problem is NP-hard for all other values of λ in the interval $[0, 1]$. We present an approximation algorithm that, for any value of the parameter λ, finds a spanning tree T of value no more than $1 + \sqrt{2}$ times larger than the value of the optimal tree.

We present an interpretation of the multiplicities of the edges in a spanning tree in terms of flows. This interpretation allows us to derive a lower bound for the optimal value of the function $f_g(T) = \sum_{e \in T} g(m(e))$. For the case when g is a convex function, we can use techniques from convex minimum cost flows to compute an integer flow F of value equal to the lower bound. By carefully re-routing fractions of the flow F, we can construct a spanning tree T for which we are able to bound its value $f_g(T)$ in terms of the lower bound.

References

- [1] G. Frederickson and R. Solis-Oba. Rooted spanning trees with small weight and average length. In *Proceedings 6th Italian Conference on Theoretical Computer Science*, pages 114–125. World Scientific Publishing, 1998.

Approximating the Achromatic Number

Investigator: Piotr Krysta

The *achromatic number problem* on a graph G is: compute the maximum size k of a vertex coloring of G, where every pair of the k colors is assigned to some pair of adjacent vertices and adjacent vertices are colored with different colors (such k is called the *achromatic number* of G). Yannakakis and Gavril [6] showed that the achromatic number problem is NP-complete. It is NP-complete also for bipartite graphs [4]. Cairnie and Edwards [1] showed that the problem is NP-complete even for trees. Cairnie and Edwards [2] have proved that the problem for trees with constant maximum degree can be solved in polynomial time. The running time of their algorithm is $\Omega(m^{126})$, where m is the number of edges of the tree.

In [5] we study combinatorial methods for approximating the achromatic number problem. Our first result is a 2-approximation algorithm for trees, which improves the 7-approximation algorithm of Chaudhary and Vishwanathan [3]. Let $d(n)$ be some (possibly increasing) function and T be a tree with n vertices. For the case that the maximum degree of T is bounded by $d(n)$, we developed an alternative to the previous, combinatorial approach to the problem. This let us reduce the approximation ratio of 2 to 1.582. An additional result is a 1.155-approximation algorithm for binary trees, *i.e.* for trees with maximum degree 3. The ratios 1.582 and 1.155 are proved to hold asymptotically as the achromatic number grows. For example, the first algorithm produces an achromatic coloring with at least $\frac{1}{1.582}\Psi(T) - O(d(n))$ colors, where $\Psi(T)$ is the achromatic number of T. We show that the algorithms for bounded degree trees can be implemented in linear time in the unit cost RAM model. Although our algorithms for bounded degree trees are approximate and the algorithm of [2] for constant degree trees is an exact one, our algorithms have linear running time and they also work on trees with larger maximum degree (*e.g.*,$d(n) = O(\log(n))$ or even for $d(n) = (n-1)^{1/4}$). Our next result is an $O(n^{3/8})$-approximation algorithm for graphs with n vertices and with *girth* (*i.e.* length of the shortest cycle) at least six. This algorithm improves the $O(n^{1/2})$-approximation in [3]. We also improve a result of Farber *et al.g* [4] giving a better lower bound for the achromatic number of binary trees.

References

[1] N. Cairnie and K.J. Edwards. Some results on the achromatic number. *Journal of Graph Theory*, 26:129–136, 1997.

[2] N. Cairnie and K.J. Edwards. The achromatic number of bounded degree trees. *Discrete Mathematics*, 188:87–97, 1998.

[3] A. Chaudhary and S. Vishwanathan. Approximation algorithms for the achromatic number. In *Proceedings 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 558–563, 1997.

[4] M. Farber, G. Hahn, P. Hell, and D. Miller. Concerning the achromatic number of graphs. *Journal of Combinatorial Theory, Series B*, 40:21–39, 1986.

- [5] P. Krysta and K. Loryś. Efficient approximation algorithms for the achromatic number. In *Proceedings 7th Annual European Symposium on Algorithms (ESA'99)*. LNCS, Springer Verlag, Berlin, 1999. To appear.

[6] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38:364–372, 1980.

Bounded Facility Location

Investigators: Piotr Krysta and Roberto Solis-Oba

The *bounded k-median problem* is to select, in an undirected graph $G = (V, E)$, a set S of k vertices (called *centers*) such that the maximum distance from a vertex $v \in V \setminus S$ to S is at most a given bound d and the average distance from vertices in $V \setminus S$ to S is minimized. Let $G = (V, E)$ be a graph with a minimum dominating set of size k. When all edge lengths are 1, and $d = 1$, the bounded k-median problem is equivalent to the minimum dominating set problem. The bounded k-median problem is also a generalization of the k-center problem. In the *bounded uncapacitated facility location problem*, given a graph $G = (V, E)$, there is a set $F \subseteq V$ of possible locations for facilities. Each vertex $i \in F$ has associated a cost f_i for opening a facility there. The cost of servicing a vertex $v \in V \setminus F$ is the distance from v to the nearest facility. The goal is to find locations for the facilities so that the total cost of servicing the vertices $v \in V \setminus F$ plus the total cost for opening facilities is minimized, and the maximum servicing cost is at most a given bound d.

In [5] we study bounded facility location problems. Let the *service cost* of a vertex be the distance from the vertex to its closest center. We prove that the bounded k-median problem is Max SNP-hard even when all edge lengths are 1, and $d = 2$. Moreover, by extending ideas of [2], we prove that the solution of the problem cannot be approximated in polynomial time within a factor smaller than 1.367 unless $NP \subseteq DTIME(n^{O(\log \log n)})$.

We present a technique for designing randomized approximation algorithms for several versions of the bounded k-median problem when all edge lengths are 1 (in these algorithms the stated performance ratio and number of centers are expected values): (i) a 1.4212-approximation algorithm for the bounded k-median problem that, with high probability, uses at most $2k$ centers and has maximum service cost $2d$, (ii) a 1.7768-approximation algorithm for the bounded k-median problem that uses at most k centers, and has maximum service cost $3d$, (iii) a 1.9321-approximation algorithm for the bounded k-median problem with maximum service cost $3d$ when the vertices have weights $\{1, +\infty\}$. For the bounded k-median problem we also give a deterministic 1.5-approximation algorithm that uses at most $2k$ centers and has service cost at most $2d$. We give also approximation algorithms for a fault tolerant bounded k-median problem: the bounded p-neighbor k-median problem. These algorithms improve, on average, the approximation ratios of the algorithms [3, 4] for

the unbounded k-center and p-neighbor k-center problems in the case of unit edge lengths. For arbitrary edge lengths, we extend algorithms of [2, 1, 7, 6] for the k-median problem, and for the capacitated and uncapacitated facility location problems. These algorithms have the same performance guarantees as the original ones, and they bound the maximum service cost of every vertex.

References

[1] F. Chudak. Improved approximation algorithms for uncapacitated facility location. In *Proceedings Integer Programming and Combinatorial Optimization (IPCO'98)*, pages 180–194. LNCS 1412, Springer Verlag, 1998.

[2] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. In *Proceedings 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 649–657, 1998.

[3] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33:533–550, 1986.

[4] S. Khuller, R. Pless, and Y.J. Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 1999. To appear.

- [5] P. Krysta and R. Solis-Oba. Approximation algorithms for bounded facility location. In *Proceedings 5th Annual International Computing and Combinatorics Conference (COCOON'99)*. LNCS, Springer Verlag, Berlin, 1999. To appear.

[6] J. H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44:245–249, 1992.

[7] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *Proceedings 29th ACM Symposium on Theory of Computing (STOC'97)*, pages 265–274, 1997.

Rooted Node-Connectivity Problems

Investigator: Zeev Nutov

Connectivity problems have important applications in areas such as network design and reliability analysis. A fundamental problem in this area is the *survivable network design problem*: Given an undirected graph with nonnegative costs on the edges, find the cheapest subgraph such that for every pair of nodes $\{i, j\}$, there are k_{ij} internally vertex disjoint $i \leftrightarrow j$-paths. Most of these problems are NP-hard, and so approximation algorithms are of interest. No efficient approximation algorithm for this problem is known, except a 3-approximation algorithm of Ravi and Williamson [5] for the case $k_{ij} \leq 2$. A particularly important case of the survivable network design problem is the problem of finding the cheapest k-connected spanning subgraph, that is the case of *uniform requirements* $k_{ij} = k$ for every $\{i, j\}$. A few approximation algorithms are known for this problem. For an arbitrary k, [5] gives a $2H(k)$-approximation algorithm, where $H(k) = 1 + \frac{1}{2} + \ldots + \frac{1}{k}$.

We consider a problem that is "sandwiched" between the two problems described above. The *multiroot k-outconnected subgraph problem* is: Given an undirected graph $\mathcal{G}$ with nonnegative costs on the edges, a vector of q root nodes $\vec{R} = (r_1, \ldots, r_q)$, and a vector $\vec{K} = (k_1, \ldots, k_q)$ of connectivity requirements, find a minimum-cost spanning subgraph such that for every $i = 1, \ldots, q$ there are k_i internally vertex disjoint paths from r_i to any other node. Let $k = \max_i k_i$ be the maximum connectivity requirement. The best known algorithm for this problem has approximation ratio $2q$, where q can be as large as $k - 1$. For no value of $k \geq 2$ was a better approximation algorithm known.

In [2], we give approximation algorithms for particular instances: (i) a 4-approximation algorithm for metric costs, and (ii) a $\min\{2, \frac{k+2q-1}{k}\}$-approximation algorithm for uniform costs. The algorithms are based on new structural results in graph connectivity. The improvements are obtained by focusing on single root k-outconnected graphs and proving: (i) an extension of a theorem by Bienstock *et al.*g [1] on splitting off edges while preserving node connectivity, and (ii) an extension of Mader's "critical cycle" theorem [3] for k-node connected graphs.

In [4], we consider the case $k_i \in \{1, 2, 3\}$, which may arise in practical networks, where connectivity requirements are usually rather small. For this case we give a (10/3)-approximation algorithm, improving the best previously known 4-approximation. Our algorithm also implies a slight improvement for arbitrary k. In the case we have an initial graph which is 2-connected, the algorithm achieves approximation ratio 2.

References

[1] D. Bienstock, E. F. Brickel, and C. L. Monma. On the structure of minimum weight k-connected spanning networks. *SIAM Journal on Discrete Mathematics*, 3:320–329, 1990.

- [2] J. Cheriyan, T. Jordán, and Z. Nutov. Approximating k-outconnected subgraph problems. In *Proceedings 1st Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)*, Lecture Notes in Computer Science, pages 77–88. LNCS 1444, Springer Verlag, 1998.

[3] W. Mader. Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen. *Archiv der Mathematik*, 23:219–224, 1972.

- [4] Z. Nutov. Approximating multiroot 3-outconnected subgraphs. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 951–952, 1999.

[5] R. Ravi and D. P. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 18:21–43, 1997.

Bin Packing with Conflicts

Investigator: Klaus Jansen

We have studied the following bin packing problem with conflicts. The input I of the problem consists of an undirected graph $G = (V, E)$ with a set of items $V = \{1, \ldots, n\}$ and sizes $s_1, \ldots, s_n$ that are associated with the items. We assume that each item size is a rational number in the interval $(0, 1]$. The edges represent conflicts between the items. The problem is to partition the set V of items into a minimum number of independent sets or bins $U_1, \ldots, U_m$ such that no conflicting items belong to the same bin and $\sum_{i \in U_j} s_i \leq 1$ for each $1 \leq j \leq m$. In other words, we want to find a conflict-free packing with a minimum number m of bins. One application of this problem is the assignment of processes to processors. In this case, we have a set of processes (*e.g.*, multimedia streams) where some of the processes are not allowed to execute on the same processor.

In [4], we have proposed several approximation algorithms A with constant absolute worst case bound $A(I) \leq \rho \cdot OPT(I)$ for the bin packing problem with conflicts for graphs that can be colored with a minimum number of colors in polynomial time. Here $OPT(I)$ denotes the optimal solution for I. Using a composed algorithm (an optimum coloring algorithm and a bin packing heuristic for each color set), we have obtained an approximation algorithm with worst case bound ρ between 2.691 and 2.7. Furthermore, using a precoloring method that works for, *e.g.*, interval graphs, split graphs and cographs, we have obtained an algorithm with bound 2.5. Based on a separation method, we have developed an algorithm with worst case ratio $2 + \epsilon$ for cographs and graphs with constant treewidth.

A d-inductive graph [2] has the property that the vertices can be assigned distinct numbers $1, \ldots, n$ in such a way that each vertex is adjacent to at most d lower numbered vertices. We assume that an order $v_1, \ldots, v_n$ is given such that $|\{v_j | j < i, \{v_j, v_i\} \in E\}| \leq d$ for each $1 \leq i \leq n$. We notice that such an order (if one exists) can be obtained in polynomial time.

The main new result in [3] is an asymptotic approximation scheme for the bin packing problem with conflicts restricted to d-inductive graphs. The time complexity of this algorithm is polynomial in n and $\frac{1}{\epsilon}$. The main ideas of the approximation scheme are the following. In the first step, we remove all items of size smaller than $\frac{\epsilon}{2}$ and apply a grouping method (of Fernandez de la Vega and Lueker [1]) to obtain a bin packing instance with a constant number of different bin sizes. Then, using the algorithm of Karmarkar and Karp [5], we generate an approximate solution for the bin packing instance without considering the conflicts. After that, we modify the generated solution such that each bin contains an independent set of items. In the last step, we insert the small items removed in the first step and use new bins only if necessary.

References

[1] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1:349 – 355, 1981.

[2] S. Irani. Coloring inductive graphs on-line. *Algorithmica*, 11:53 – 72, 1994.

• [3] K. Jansen. An approximation algorithm for bin packing with conflicts. In *Proceedings Skandinavian Workshop on Algorithm Theory (SWAT'98)*, pages 35 – 46. LNCS 1432, Springer Verlag, 1998.

[4] K. Jansen and S. Öhring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132:85 – 108, 1997.

[5] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin packing problem. In *Proceedings Symposium on the Foundations of Computer Science (FOCS'82)*, pages 312 – 320, 1982.

Algorithms for Optical Networks

Investigator: Klaus Jansen

In optical communication networks, a *connection request* in a directed tree T is given by a sender-receiver pair (u, w) and corresponds to the directed path from u to w in T. Two paths *intersect* if they share a directed edge of T. The *load* $L(e)$ of a directed edge e of T is the number of paths in a given set P of paths using edge e, and L denotes the maximum load among all edges of T. A W-coloring of a given set of paths is an assignment of colors (wavelengths) to the paths using at most W colors such that intersecting paths receive different colors. For a given directed tree T, set P of paths in T, and number W of available colors, the *maximum path coloring* problem (MaxPC) is to compute a subset $P' \subseteq P$ and a W-coloring of P' such that $|P'|$ is maximized.

Previous work has focused on the *path coloring problem*, where the goal is to assign wavelengths to all given connection requests while minimizing the number of different wavelengths used. For undirected trees, a (3/2)-approximation algorithm was given in [6] and improved to an asymptotic 1.1-approximation in [2]. For directed trees, the best known algorithm colors a given set of directed paths with maximum load L using at most $\lceil (5/3)L \rceil$ colors [5, 1]. We gave an efficient implementation and improvement of the running time of the approximation algorithm for this problem in [3]. While the path coloring problem is relevant when a provider designs a network in order to meet the given demands or when the network has enough capacity for satisfying all given requests, MaxPC

applies to the case where an existing network has insufficient capacity and the goal is to maximize the number of accepted requests.

If $W = 1$, *i.e.*, only one wavelength is available, then MaxPC is equivalent to finding a maximum cardinality subset of edge-disjoint paths. We give in [4], as our main result, a family of polynomial-time approximation algorithms with approximation ratio $5/3 + \varepsilon$ for this case, where ε can be chosen arbitrarily small. For MaxPC with arbitrary W, we obtain a 2.22-approximation for trees of arbitrary degree and a 1.58-approximation for trees whose degree is bounded by a constant.

References

[1] T. Erlebach C. Kaklamanis, P. Persiano and K. Jansen. Constrained bipartite edge coloring with applications to wavelength routing. In *Proceedings 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, pages 493–504. LNCS 1256, Springer Verlag, 1997.

[2] T. Erlebach and K. Jansen. Scheduling of virtual connections in fast networks. In *Proceedings 4th Parallel Systems and Algorithms Workshop (PASA'96)*, pages 13–32. World Scientific Publishing, 1997.

- [3] T. Erlebach and K. Jansen. Efficient implementation of an optimal greedy algorithm for wavelength assignment. In *Proceedings of Workshop on Algorithm Engineering (WAE'98)*, pages 13 – 24, 1998.

- [4] T. Erlebach and K. Jansen. Maximizing the number of connections in optical tree networks. In *Proceedings International Symposium on Algorithms and Computation (ISAAC'98)*, pages 179 – 188. LNCS 1533, Springer Verlag, 1998.

[5] C. Kaklamanis K. Jansen, T. Erlebach and P. Persiano. An optimal greedy algorithm for wavelength allocation in directed tree networks. In *Proceedings DIMACS Workshop on Network Design: Connectivity and Facilities Location (1997)*, volume 40, pages 117–129. Dimacs Series in Discrete Mathematics and Theoretical Computer Science, 1997.

[6] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proceedings 26th Annual ACM Symposium on Theory of Computing (STOC'94)*, pages 134–143, 1994.

Admission Control

Investigator: Stefano Leonardi

In [1] we study the multicast routing and admission control problem on unit-capacity tree and mesh topologies in the throughput maximization model. Every multicast group active in the network is associated with a tree where a specific node is designated as the source of the multicast group. Every group receives from a subset of the nodes of the network a set of requests for joining the group. The algorithm must design a multicast tree for every group that spans the source and a number of request nodes of the group. The goal is to maximize the number of requests that are satisfied, under the constraint that the trees of different groups must be edge-disjoint, *i.e.* every connection request asks for the whole bandwidth available over the links of the network. Polynomial time approximation algorithms are presented for the off-line version of the problem. An approximation algorithm with constant approximation ratio based on a greedy technique is presented for tree networks, while for mesh network topologies a $poly(\log \log n)$ approximation algorithm using a fractional LP formulation of the problem and randomized rounding techiniques is given. In the online version of the problem, requests to join a multicast group are presented one by one while the multicast trees are incrementally constructed. For the on-line version of the problem, randomized algorithms with polylogarithmic competitive ratios are presented. These are the first randomized on-line algorithms on tree and mesh topologies for the case in which connection requests ask for the whole link bandwidth.

References

- [1] M. R. Henzinger and S. Leonardi. Scheduling multicasts on unit capacity trees and meshes. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 438–447, 1999.

3.2 Online Algorithms

Delay in Online Algorithms

Investigator: Susanne Albers

Most on-line analysis assumes that, at each time-step, all relevant information up to that time step is available and a decision has an immediate effect. In many on-line problems, however, the time relevant information is available and the time a decision has an effect may be decoupled. For example, when making an investment, one might not have completely up-to-date information on market prices. Similarly, a buy or sell order might be executed only some time later in the future. Related timing problems occur when a *group* of people or agents makes decisions. The group might come together only at particular time instances. The actions are delayed, in that they can only occur at specific points in time.

The importance of when information becomes available has been noted previously, especially in the significant body of work on algorithms with lookahead, see, *e.g.*, [3, 4]. In the case of on-line decision models, however, the possibility of not having up-to-date information is not generally addressed. For load balancing problems, the question has been considered for statistical models [5]. There is also a large body of work on algorithms with distributed agents, who must coordinate their efforts in the face of some cost for communication, see, *e.g.*, [2, 6]. These models, however, model communication as an instantaneous event, and hence the communication cost does not directly incorporate a notion of time and delay.

In [1], we consider several standard on-line problems and examine their generalizations to delayed models. These generalizations are quite natural and lead to interesting insight into the original problem. First, we study the delayed information model applied to the classical problem of on-line scheduling on parallel machines to minimize the makespan. Here a scheduling algorithm must assign new jobs to processors based on stale load information. Traditional algorithms for on-line scheduling do poorly in this scenario. We develop new algorithms for this model and prove almost matching lower bounds. Furthermore, we study the list update problem in the delayed action model and prove nearly tight upper and lower bounds for deterministic on-line algorithms. We also show that a randomized on-line algorithm can only beat the deterministic lower bound if it uses paid exchanges. Next, we generalize an on-line stock market model by studying natural delayed models. Finally, we apply the delayed action model to the general class of relaxed metrical task systems. Relaxed task systems are an abstract model for problems where one has to decide when it is appropriate to make expensive configuration changes. This class includes the ski rental problem, page migration file replication, network leasing, and other problems. We extend known results to apply to relaxed task systems with delayed action, effectively handling the delayed models of an entire general class of problems.

References

- [1] S. Albers, M. Charikar, and M. Mitzenmacher. On delayed information and action in on-line algorithms. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 71–80, 1998.

[2] N. Alon, G. Kalai, M. Ricklin, and L. Stockmeyer. Lower bounds on the competitive ratio for mobile user tracking and distributed job scheduling. In *Proceedings 33rd Annual Symposium on the Foundations of Computer Science (FOCS'92)*, pages 334–343, 1992.

[3] S. Ben-David and A. Borodin. A new measure for the study of on-line algorithms. *Algorithmica*, 11:73–91, 1994.

[4] S. Irani. Coloring inductive graphs on-line. *Algorithmica*, 11:53–62, 1994.

[5] M. Mitzenmacher. How useful is old information? In *Proceedings 16th Annual ACM Symposium on Principles of Distributed Computing (PODC'97)*, pages 83–91, 1997.

[6] C. H. Papadimitriou and M. Yannakakis. On the value of information in distributed decision making. In *Proceedings 25th ACM Symposium on Principles of Distributed Computing (PODC'91)*, pages 61–64, 1991.

On the Bahncard Problem

Investigator: Rudolf Fleischer

The Bahncard is a railway pass of the Deutsche Bundesbahn (the German railway company) that entitles its holder to a 50% price reduction on nearly all train tickets. It costs 240 DM, and it is valid for 12 months. Similar bus or railway passes can be found in many other countries.

For the common traveler, the decision at which time to buy a Bahncard is a typical online problem, because she usually does not know when and to which place she will travel next. Note that the *Bahncard Problem* is a generalization of the Ski Rental Problem [2, p. 113]. In [1] we show that the greedy algorithm applied by most travelers and clerks at ticket offices is not better in the worst case than the trivial 2-competitive algorithm that never buys a Bahncard. We present two optimal $\frac{3}{2}$-competitive algorithms, an optimistic one and and a pessimistic one. The randomized versions of these algorithms are $\frac{4}{3}$-competitive against any oblivious adversary. We give a lower bound of $\frac{e}{e-\frac{1}{2}}$ for randomized online algorithms and present an algorithm that we conjecture to be optimal; the conjecture is proven for a special case of the problem.

References

- [1] R. Fleischer. On the Bahncard problem. In *Proceedings 4th Annual International Computing and Combinatorics Conference (COCOON'98)*, pages 65–74. LNCS 1449, Springer Verlag, 1998. To appear in *Theoretical Computer Science*.

[2] P. Raghavan. Lecture notes on randomized algorithms. Technical Report RC 15340 1/9/90, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York, 1990.

Randomized Scheduling on Identical Machines

Investigator: Steve Seiden

The problem of scheduling jobs on m identical machines is considered to be a classic one in computer science. We investigate the version of this problem where jobs are given to the scheduler one by one. Each job must be assigned to a machine before the next job is revealed. The duration of each job is fixed, and each job is performed equally well on any machine. The goal is to minimize the makespan—the last completion time of a job.

The study of this problem was initiated in 1966 by Graham who developed an algorithm called LIST which is $(2 - \frac{1}{m})$-competitive [4]. Since Graham's seminal work, many researchers have

investigated this problem. The deterministic competitive ratio is known to lie in the interval $[1.852, 1.923]$ for large m [1].

Although much attention has been given to this problem, almost all of it has focused on deterministic algorithms. The exceptions are as follows: For $m = 2$ an algorithm achieving a competitive ratio of $\frac{4}{3}$ is presented by Bartal, Fiat, Karloff and Vohra [2]. They also show a matching lower bound for two machines. In [7] we presented an algorithm for $m \geq 3$. Epstein, Noga, Seiden, Sgall and Woeginger have given much simpler randomized algorithms for $m = 2, 3$ [3]. However, their algorithms trade performance for simplicity.

A barely random algorithm is one which uses a distribution over a constant number of deterministic strategies [5]. Barely random algorithms are desirable in that they conserve a precious resource—random bits. Neither of the algorithms given in [2, 7] is barely random. In fact, both of these algorithms potentially make a random choice for each job scheduled. Further, both algorithms use $\Omega(nm)$ variables, and use a total of $\Omega(n^2 \log m + m)$ time, where n is the length of the job sequence. In [6], we present an adaptation of the algorithm of [7]. Our algorithms use only $O(m)$ variables, and take a total of $O(n \log m + m)$ time. They each use only 11 random bits, with only a slight degradation in the competitive ratio. Further, the changes make the analysis simpler than that of [7].

References

[1] S. Albers. Better bounds for online scheduling. In *Proceedings 29th ACM Symposium on Theory of Computing (STOC'97)*, pages 130–139, 1997.

[2] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra. New algorithms for an ancient scheduling problem. *Journal of Computer and System Sciences*, 51(3):359–366, Dec 1995.

[3] L. Epstein, J. Noga, S. S. Seiden, J. Sgall, and G. Woeginger. Randomized online scheduling on two uniform machines. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 317–326, Jan 1999.

[4] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell Systems Technical Journal*, 45:1563–1581, 1966.

[5] N. Reingold, J. Westbrook, and D.D. Sleator. Randomized competitive algorithms for the list update problem. *Algorithmica*, 11(1):15–32, Jan 1994.

- [6] S. S. Seiden. Barely random algorithms for multiprocessor scheduling. Submitted for publication.

[7] S. S. Seiden. Randomized online multiprocessor scheduling. *Algorithmica*, 1999. To appear.

Scheduling with Rejection

Investigator: Steve Seiden

In a variant of the classic scheduling problem just discussed, jobs may also be rejected. This means that the algorithm need only schedule a subset of the given jobs. However, this rejection does not come for free. The algorithm pays a penalty for each rejected job. The goal is to minimize the sum of the makespan and rejection penalties. This problem was proposed and studied in [1]. In [2], we present a 1.44127-competitive randomized algorithm for two machines.

It is also interesting to look at preemptive scheduling with rejection. When preemption is allowed, a job may be split among machines in a restricted way. We introduced this problem in [4] and gave a deterministic upper bound of 2.38743. We have recently managed to prove the first non-trivial lower bound for the problem—the competitive ratio of any online algorithm is at least

2.12457. We also have shown lower bounds for certain classes of algorithms. These results have been included into the journal submitted version [3].

References

[1] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, and L. Stougie. Multiprocessor scheduling with rejection. In *Proceedings Seventh ACM-SIAM Symposium on Discrete Algorithms*, pages 95–103, Jan 1996.

- [2] S. S. Seiden. A manifesto for the computational method. Submitted for publication, preliminary version appeared in *Proceedings of Fun with Algorithms*, June 1998.
- [3] S. S. Seiden. Preemptive multiprocessor scheduling with rejection. Submitted for publication.

[4] S. S. Seiden. More multiprocessor scheduling with rejection. Technical Report Woe-16, TU Graz, Institut für Mathematik B, 1997.

Scheduling with Machine Breakdowns

Investigator: Susanne Albers

In [1] we investigate an online version of a basic scheduling problem where a set of jobs has to be scheduled on a number of identical machines so as to minimize the makespan. The job processing times are known in advance and preemption of jobs is allowed. Machines are *non-continuously* available, *i.e.*, they can break down and recover at arbitrary time instances *not known in advance.* New machines may be added as well. Thus machine availabilities change online.

We first show that no online algorithm can construct optimal schedules. We also show that no online algorithm can achieve a bounded competitive ratio if there may be time intervals where no machine is available. Then we present an online algorithm that constructs schedules with an optimal makespan of $C_{\max}^{OPT}$ if a *lookahead* of one is given, *i.e.*, the algorithm always knows the next point in time when the set of available machines changes. Finally, we give an online algorithm without lookahead that constructs schedules with a nearly optimal makespan of $C_{\max}^{OPT} + \epsilon$, for any $\epsilon > 0$, if at any time at least one machine is available. Our results demonstrate that not knowing machine availabilities in advance is of little harm.

References

- [1] S. Albers and G. Schmidt. Scheduling with unexpected machine breakdowns. In *Proceedings of APPROX'99,* To appear. LNCS, Springer Verlag, 1999.

Robot Exploration

Investigators: Susanne Albers and Klaus Kursawe

In [1] we continued our work on *exploration problems*, where a robot has to construct a complete map of an unknown environment using a path that is as short as possible.

First, we consider a geometric setting. The robot is placed in a room with obstacles, which are modeled by simple polygons. The robot has 360° vision and can see an infinite range as long as no obstacle or exterior wall blocks the view. Deng, Kameda and Papadimitriou [3] developed an $O(n)$-competitive algorithm for exploring rectilinear rooms with n rectilinear obstacles. They conjectured that there exists an algorithm that achieves a constant competitive ratio. In [1] we have disproven this conjecture: We show that no deterministic or randomized online exploration algorithm in scenes with n rectangular obstacles can be better than $\Omega(\sqrt{n})$-competitive. We can extend our

bound to three-dimensional scenes without obstacles. We show that no algorithm for exploring the interior of a rectilinear polyhedron with n vertices can be better than $\Omega(\sqrt{n})$-competitive.

Second, we consider a graph theoretic setting. In practice, the robot's sensors cannot scan an infinite range but can scan only a few meters. This constraint can be modeled by adding a grid to the scene and requiring that the robot moves on the nodes and vertices of the grid. A node in the grid models the vicinity that the robot can see at a given point. Now the robot has to explore all nodes and edges of the grid using as few edge traversals as possible. Betke, Rivest and Singh [2] introduced an interesting, more complicated variant of this problem where an additional *piecemeal constraint* has to be satisfied, *i.e.*, the robot has to return to a start node s every so often. These returns might be necessary because the robot has to refuel or drop samples collected on a trip. Betke *et al.* gave an efficient algorithm for exploring grids with rectangular obstacles. We [1] present an efficient strategy for piecemeal exploration of grids with arbitrary obstacles.

References

- [1] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 842–843, Baltimore, USA, 1999.

[2] M. Betke, R. Rivest, and M. Singh. Piecemeal learning of an unknown environment. In *Proceedings 5th Conference on Computational Learning Theory (COLT'93)*, pages 277–286, 1993.

[3] X. Deng, T. Kameda, and C. H. Papadimitriou. How to learn an unknown environment. *Journal of the ACM*, 45:215–245, 1998.

Page Replication

Investigators: Rudolf Fleischer and Steve Seiden

In the online page replication problem, one must decide which nodes of a given network should have a copy of a fixed database (or page). Initially only one node has this page. The page can be replicated to any other node, but only at high cost. Users of the network make requests for data at their respective nodes. How do we best serve these requests? If the page is located close by, the requests can be easily served, but if not we may want to copy the page to a location close to where we are currently receiving requests.

This problem was introduced by Black and Sleator [2], and further studied by Albers and Koga [1], and Głazek [4]. Formally, it can be stated as follows: We have a fixed finite metric M with a distinguished origin vertex s. A sequence of requests is given. Before serving each request, the online algorithm has the option of duplicating the page. In other words, it can copy the database from any current vertex x that has it, to another vertex y that does not at cost $D \cdot d(x, y)$ where D is the replication factor and $d(x, y)$ is the distance between x and y. Then the request is served at cost $d(r, x)$ where r is the request vertex and x is the closest vertex that has the page. Most research on this problem has focused on two types of metric spaces: trees and rings. In [3] we present several new results about this problem: (1) In their seminal paper about the page replication problem, Black and Sleator [2] claim that 2.5 is a lower bound for "the four node cycle", but give no proof. We give evidence against their claim, showing an upper bound of 2.36603 for the ring with four evenly spaced nodes. (2) We give two new lower bounds for rings: a deterministic lower bound of 2.31023 and a randomized lower bound of 1.75037. (3) We initiate the study of replication in continuous metric spaces. Deterministic algorithms in continuous metric spaces correspond naturally to randomized algorithms in discrete metric spaces with the same competitive ratio. We give a 1.58198-competitive deterministic algorithm for continuous trees and

a 2.54150-competitive deterministic algorithm for continuous rings. These algorithms correspond to the randomized discrete algorithms by Albers and Koga [1] and Głazek [4], respectively, but our proofs are much simpler. (4) We investigate a randomized algorithm for the ring proposed by Albers and Koga [1]. They showed that this algorithm is 3.16396-competitive. We present a modification of their algorithm which is 2.37297-competitive. This is the best known upper bound for the ring.

References

[1] S. Albers and H. Koga. New on-line algorithms for the page replication problem. *Journal of Algorithms*, 27(1):75–96, Apr 1998.

[2] D. L. Black and D. D. Sleator. Competitive algorithms for replication and migration problems. Technical Report CMU-CS-89-201, Department of Computer Science, Carnegie-Mellon University, 1989.

- [3] R. Fleischer and S. S. Seiden. Page replication—Variations on a theme. Manuscript. Presented at the *International Conference on Optimization (SIGOPT'99)*, Trier, Germany, March 1999.

[4] W. Głazek. On-line algorithms for page replication in rings. Presented at the *ALCOM Workshop on On-line Algorithms*, Udine, Italy, Sep 1998.

Online Call Admission

Investigator: Stefano Leonardi

In [1] we consider the problem of on-line call admission and routing on tree and mesh network topologies. A set of communication requests between pairs of vertices in the network is presented online. Every time a new communication request is presented, the algorithm has to decide whether to accept or to reject the request. Accepted requests must be routed on a virtual circuit in the network under link bandwidth constraints. This paper presents randomized algorithms that obtain almost optimal expected competitive ratios while featuring a narrow distribution around the expectation. This work introduces the notion of trade-off between competitive ratio and variance in a randomized on-line algorithm. Previous work on these problems provides randomized algorithms that can obtain almost optimal expected competitive ratio at the cost of a high variance.

References

- [1] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosén. On-line randomized call-control revisited. In *Proceedings 9th ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 323–332, 1998.

Bin Packing

Investigator: Susanne Albers

In our previous progress report, we described some initial work on online bin packing. Since then, we have extended this work and published it in [1]. We briefely sketch the results again. In [1] we prove that the First Fit bin packing algorithm is stable under the input distribution $U\{k-2,k\}$ for all $k \geq 3$, settling an open question from the recent survey by Coffman, Garey, and Johnson [2]. Our proof generalizes the multi-dimensional Markov chain analysis used by Kenyon, Rabani, and Sinclair to prove that Best Fit is also stable under these distributions [3]. Our proof is motivated by an analysis of Random Fit, a new simple packing algorithm related to First Fit, that is interesting in its own right. We show that Random Fit is stable under the input distributions $U\{k-2,k\}$, as

well as present worst-case bounds and some results on distributions $U\{k-1,k\}$ and $U\{k,k\}$ for Random Fit.

References

- [1] S. Albers and M. Mitzenmacher. Average case analyses of first-fit and random-fit bin packing. In *Proceedings 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 290–299, 1998.

  [2] E.G. Coffman Jr., M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing: A survey. In *Approximation Algorithms for NP-hard Problems,* edited by D. Hochbaum, PWS, 1996.

  [3] C. Kenyon, A. Sinclair, and Y. Rabani. Biased random walks, Lyapunov functions, and stochastic analysis of best fit bin packing. In *Proceedings 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95)*, pages 351–358, 1995.

Surveys

Investigators: Susanne Albers and Stefano Leonardi

Online algorithms have received a lot of research interest during the last ten years. The first book on online algorithms [2] and a number of survey articles appeared last year. Recently, we wrote a short survey summarizing some extensively studied application areas and outlining some important trends and directions for future research [1]. We also presented a survey of the work done during the last years on on-line network routing problems [3].

References

- [1] S. Albers and S. Leonardi. Online algorithms. *ACM Computing Surverys, Electronic Issue*, 1999.

  [2] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

- [3] S. Leonardi. On-line network routing. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 242–267. LNCS 1442, Springer Verlag, 1998.

4 Parallel and External Computing

In our group there has always been substantial work on models of computation that go beyond the simple sequential RAM model with uniform memory access cost. However, our effort in this area has been refocused due to changes in our staff and due to shifting points of view in our group and in the community as a whole. Distributed computing is underrepresented since Marina Papatriantafilou and Philippas Tsigas have left. Whereas PRAMs were once considered *the* mainstream model of parallel computing, we now often view a PRAM algorithm as a first (important) step before moving on to tackling the problem for a less optimistic computational model and to actual implementation. Examples of this development are our work on shortest paths and on list ranking in Section 4.1.4. Also, one of our larger PRAM projects has been completed now [2, 3].

We begin our summary of work on parallel processing with fundamental techniques for communication in interconnection networks (Section 4.1.1) and for load balancing (Section 4.1.2). As before, the core of our work are solutions of classical algorithmic problems like sorting (Section 4.1.3) and, in particular, graph problems such as shortest paths, list ranking or spanning trees (Section 4.1.4). A few further results can be found in Section 2.1 (heap construction) and in Section 5.1. We also have cooperations involving applications of parallel computing in chemistry (Section 6.3.1), physics [4] and operations research [1, 5].

The model of *external computing* has gained so much impetus in the last two years that we dedicate about half of this section to it. One reason is that, in spite of the ever-increasing sizes of internal memories, data sets arising in important applications like geographic information systems, computer graphics, WWW-search, data warehouses, electronic libraries, or scientific computing, are still too large to fit in memory. Archives for satellite images, climate simulation or elementary particle physics already work with petabytes of data. For many of these applications, no size limits are in sight. Most of our work uses the model where one data block of size B can be exchanged between an internal memory of size M and the external memory in one I/O step [6]. This simple model is already sufficient to express the two main issues in external computing: Access latencies are 10^4 to 10^6 times slower than internal accesses, and data is moved in large chunks to achieve sufficient bandwidth. Many internal algorithm that do not honor these restrictions therefore become hundreds or thousands of times slower when the system starts swapping pages from external memory.

Since sorting was the first intensively studied problem in external computing, we begin our survey with several variants of this problem. The selection problem is considered in Section 4.2.1. Priority queue data structures (Section 4.2.2) are a generalization of sorting that are important for several optimization and simulation purposes. The text-indexing data structures covered in Section 4.2.3 are central for many text processing tasks and have numerous applications. One application in computational biology is described in Section 6.1.2.

In Section 4.2.4, two interesting results on external geometric computing are described. A dynamic point location problem can be solved using a B-tree-like data structure. A large class of problems can be solved using the technique of randomized incremental construction. For trapezoidal decomposition and several related problems, this yields the first I/O-optimal algorithm.

Another reason why the external memory model is interesting is that even applications that fit into main memory are affected by the hierarchy of caches present in most modern computers. In Section 4.2.5 we give both theoretical and experimental evidence that external memory algorithms can often turn out to be also the best algorithms for main memory.

Last but not least, Section 4.2.6 summarizes work on a LEDA extension for secondary memory that simplifies and bundles experimental research and helps to accelerate the transfer of academic

results into applications.

References

- [1] P. Alefragis, C. Goumopoulos, E. Housos, P. Sanders, T. Takkula, and D. Wedelin. Parallel crew scheduling in PAROS. In *4th Euro-Par*, number 1470 in LNCS, pages 1104–1113. Springer, 1998.
- [2] Jörg K., C. W. Keßler, and J. L. Träff. *Practical PRAM Programming.* Wiley International, 1999. Book in preparation, scheduled for late 1999.
- [3] C. W. Keßler and J. L. Träff. Language and library support for practical PRAM programming. *Parallel Computing*, 1998. To appear.
- [4] U. Rathe, P. Sanders, and P. Knight. A case study in scalability: an ADI method for the two-dimensional time-dependent Dirac equation. *Parallel Computing*, 1999. To appear.
- [5] P. Sanders, T. Takkula, and D. Wedelin. High performance integer optimization for crew scheduling. In *7th International Conference on High Performance Computing and Networking Europe*, number 1593 in LNCS, pages 3–12, 1999. To appear.

[6] J. S. Vitter and E. A. M. Shriver. Algorithms for parallel memory I: Two level memories. *Algorithmica*, 12(2–3):110–147, 1994.

4.1 Parallel Computing

4.1.1 Algorithms for Interconnection Networks

Investigators: Ulrich Meyer, Peter Sanders, and Jop F. Sibeyn

Most existing parallel computers have a distributed memory. On such a parallel computer, the processors must communicate through a rather sparse interconnection network, to access data that they do not hold themselves. Therefore, the performance of a parallel computation depends to a large extent on the efficiency of the applied communication algorithms. Building upon research from the previous years, we have obtained some further results.

We have been looking at the standard mesh model in which n^2 processing units (PUs) are arranged in a square $n \times n$ grid. By refined deterministic techniques we were able to match the performance of earlier randomized algorithms for communication and sorting [10, 12].

For the elementary communication problem in which each PU is the source and destination of exactly one packet, we have obtained an algorithm that is strictly optimal with respect to the number of steps, and at the same time never stores more than a small constant number of packets in any of the PUs [15].

For the more general problem that every PU is the source and destination of k packets, $k \cdot n/2$ steps is a lower bound. This bound can be closely approximated by a simple randomized algorithm that also works for tori (meshes in which every node is connected to the same numbe of other nodes) [4].

For performing the routing operation under which all packets must be routed as in a transpose of a matrix, we have achieved optimality for two-dimensional meshes and presented novel ideas to perform transposes on higher-dimensional meshes. Furthermore, we show how to speed up sorting algorithms by using transposes. For the two-dimensional case, we have also investigated the extent to which the theoretical improvements make sense in practice [2].

In addition to the more common packet routing algorithms, we have been studying collective operations. In the gossiping problem, all p processors initially know a certain amount of information of size s, which must be routed such that in the end, all PUs have the complete set of information

of size $p \cdot s$. We considered the problem on meshes and tori of arbitrary dimensions, under both the wormhole model [14] and the store-and-forward model [5, 6]. In the latter case our algorithms are time-independent, *i.e.*, in all steps, the processors forward the received packets in the same way.

In an attempt to become more practical we have incorporated the so-called "start-up time" into the routing model. For the case that the time is either fully dominated by this start-up time or by the time needed to transfer packets, many operations are rather simple. For the intermediate case, however, clever schedules perform better than simple ones, even in practice [9].

In addition to to this standard model, we have been looking at variants. A natural generalization is that one looks at grids that are not composed of squares, but of triangles or hexagons. It turns out that, if one takes the the smaller or larger degree of the nodes into account, these networks are just as effective for routing as the square grids [8].

Another possible variant is that one assumes that in addition to the standard connections, which give the possibility to communicate with the neighbors, there are some additional connections that allow for communication over larger distances. Such connections are called buses. Particularly for problems that rely on the fast dissemination of some sparse information, these networks are very powerful, but one can also design very interesting routing algorithms for them [3].

The whole network can also be composed just of a set of buses. So, there are no normal connections, only buses. An interesting possibility is to reduce the number of PUs to $O(n)$. These may be situated on the outside [1] or on the diagonal [11]. In the latter case basic operations can be performed highly efficiently. In addition, the concept of diagonal has a pretty generalization for higher dimensions.

In [13], we provide an overview of much of the literature in this area that has appeared during the past several, very fruitful years. In the future we want to put more emphasis on actual implementation. As one preparatory step, we have designed a detailed communication benchmark for the communication library MPI (Message Passing Interface) [7]. It turns out that even vendor libraries for expensive parallel machines often use rather crude algorithms, so that there is a large potential for useful technology transfer.

References

- [1] B. S. Chlebus, A. Czumaj, and J. F. Sibeyn. Routing on the PADAM: Degrees of optimality. In C. Lengauer, M. Griebl, and S. Gorlatch, editors, *Proceedings of the 3rd International Euro-Par Conference on Parallel Processing (Euro-Par-97)*, volume 1300 of *Lecture Notes in Computer Science*, pages 272–279, Passau, Germany, 1997. Springer.
- [2] M. Kaufmann, U. Meyer, and J. F. Sibeyn. Matrix transpose on meshes: Theory and practice. *Computers and Artificial Intelligence*, 16(2):107–140, May 1997.
- [3] M. Kaufmann, R. Raman, and J. F. Sibeyn. Routing on meshes with buses. *Algorithmica*, 18(3):417–444, 1997.

  [4] M. Kaufmann and J. F. Sibeyn. Randomized multipacket routing and sorting on meshes. *Algorithmica*, 17(3):224–244, 1997.
- [5] U. Meyer and J. F. Sibeyn. Gossiping large packets on full-port tori. In *Proc. 4th International Euro-Par Conference (Euro-Par-98)*, volume 1470 of *Lecture Notes in Computer Science*, pages 1040–1046, Southampton, United Kingdom, September 1998. Springer.
- [6] U. Meyer and J. F. Sibeyn. Gossiping large packets on full-port tori. Technical report, Max-Planck-Institut für Informatik (MPII), Saarbrüecken, 1998.
- [7] R. Reussner, P. Sanders, L. Prechelt, and M. Müller. SkaMPI: A detailed, accurate MPI benchmark. In *EuroPVM/MPI*, number 1497 in LNCS, pages 52–59, 1998.

- [8] J. F. Sibeyn. Routing on triangles, tori and honeycombs. *International Journal on the Foundations of Computer Science*, 8(3):269–287, 1997.
- [9] J. F. Sibeyn. Routing with finite speeds of memory and network. In *Proceedings of the 22nd Symposium on the Mathematical Foundations of Computer Science (MFCS-97)*, volume 1295 of *Lecture Notes in Computer Science*, pages 488–497, Bratislava, Slovakia, 1997. Springer.
- [10] J. F. Sibeyn. Sample sort on meshes. In C. Lengauer, M. Griebl, and S. Gorlatch, editors, *Proceedings of the 3rd International Euro-Par Conference on Parallel Processing (Euro-Par-97)*, volume 1300 of *Lecture Notes in Computer Science*, pages 389–398, Passau, Germany, 1997. Springer.
- [11] J. F. Sibeyn. Solving fundamental problems on sparse-meshes. In Stefan Arnborg and Lars Ivansson, editors, *Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (SWAT-98)*, volume 1432 of *Lecture Notes in Computer Science*, pages 288–299, Stockholm, July 1998. Springer.
- [12] J. F. Sibeyn. Better deterministic routing on meshes. In *Proc. 13th International Parallel Processing Symposium*. IEEE Computer Society Press, 1999.
- [13] J. F. Sibeyn, M. D. Grammatikakis, D. F.Hsu, and M. Kraetzl. Packet routing in fixed-connection networks: A survey. *Journal of Parallel and Distributed Computing*, 54:77–132, 1998.
- [14] J. F. Sibeyn, B. Juurlink, and P. S. Rao. Gossiping on meshes and tori. *IEEE Transactions on Parallel and Distributed Systems*, 9(6):513–525, June 1998.
- [15] J. F. Sibeyn, M. Kaufmann, and B. S. Chlebus. Deterministic permutation routing on meshes. *Journal of Algorithms*, 22(1):111–141, 1997.

4.1.2 Dynamic Loop Scheduling

Investigator: Hannah Bast

A rich source of parallelism in compute-intensive code are *parallel loops*, that is, loops whose iterates may be executed in any order. We consider the problem of scheduling parallel loops on a given number of identical processors so as to minimize the overall execution time (makespan). We assume that the iterates are maintained by a *central scheduler* that can assign chunks of an arbitrary number of iterates to idle processors. The difficulty is that, due to either algorithmic or systemic effects, the processing time required for a chunk is not entirely predictable—usually less so for larger chunks—and that processors suffer a fixed delay for each assignment. In such a setting, it turns out that in order to achieve minimal overall finishing time, a scheduling strategy must actually minimize the *wasted time* of the schedule, that is, the sum of all delays plus the idle times of processors waiting for the last processor to finish.

This problem has been the focus of considerable research interest (*e.g.*, [6, 4, 7, 5]), but a thorough theoretical understanding has clearly been missing. Indeed, all of the more sophisticated existing strategies are purely heuristic, and it seemed to be, above all, a matter of luck whether or not a particular design principle led to a good performance. So, for example, the quite involved *factoring* (FAC) scheme of [4] tends to perform rather poorly under certain circumstances, while its quick-and-dirty variant FAC2, which does away with all the complicated heuristics, almost always yields better results and turned out to be a generally sound scheme. A comparison of many existing heuristic schemes is provided in the experimental study of [5]; we reported on this in the last progress report. The study introduces yet another scheduling strategy, which in the considered simulations outperforms all other schemes. Due to its extreme complexity, however, again no rigorous analysis supporting the experimental findings is provided.

Addressing this deficit, we propose in [1] a very general mathematical framework for the parallel-loop scheduling problem together with a complete analysis. Our model is based on estimated ranges

$[\,\alpha(w), \beta(w)\,]$ for processing times of chunks of size w, and on a measure ϵ for the overall deviation of actual processing times from these estimates, the latter not being revealed to the scheduler until after the event. For arbitrary combinations of the parameters α, β, and ϵ, we derive an upper bound for the wasted time incurred by the new *balancing* strategy. Via a matching lower bound, we also demonstrate that no strategy can do better. This very general deterministic result immediately implies upper bounds for a whole variety of probabilistic settings, where iterate execution times are assumed to be random variables. In fact, one very specialized such setting, where iterate execution times are assumed to be independent, identically distributed random variables, underlies most of the aforementioned previous work. Under these assumptions, we can show through appropriate tail estimates that for n iterates and p processors, the balancing strategy achieves an expected wasted time of $O(\log\log(n/p))$. We also establish a matching lower bound, which, in particular, settles a conjecture put forward in [5].

While [1] is primarily concerned with finding the theoretically optimal solutions for any conceivable setting, the emphasis of [2] is more on practicability. Namely, in [2] we investigate the class of the very simple so-called *linear self-scheduling* (LSS) schemes, which are in fact specializations of the balancing scheme considered in [1]. The defining characteristic of an LSS scheme is that chunk sizes decrease geometrically, that is, the size of each chunk is just one Cth of that of its predecessor, for some fixed constant $C > 1$. For this class of scheduling strategies, we are able to provide an exact analysis and determine the optimal value of C for a variety of settings. As a by-product, we are able to prove sharp bounds on the wasted time incurred by the aforementioned FAC2 scheme of [4], thus providing, for the first time, theoretical evidence for its relative success as a load-balancing scheme in various applications. However, we also show that for an appropriate choice of C, LSS outperforms FAC2 by a factor of 2 in terms of the incurred wasted time. The conclusion we draw in [2] is that if the underlying platform allows for the implementation of a scheduling scheme with decreasing chunk sizes, it should be of the LSS type.

References

- [1] H. Bast. Dynamic scheduling with incomplete information. In *Proc. 10th Annual Symposium on Parallel Algorithms and Architectures (SPAA'98)*, pages 182–191, 1998.
- [2] H. Bast. Parallel-loop scheduling made easy, manuscript. To appear as a technical report, 1999.
- [3] H. Bast. Provably optimal scheduling of irregular parallel loops. Ph.D. thesis, in preparation, 1999.

[4] L. E. Flynn, S. F. Hummel, and E. Schonberg. Factoring: A method for scheduling parallel loops. *Comm. Assoc. Comput. Mach.*, 35:8:90–101, 1992.

- [5] T. Hagerup. Allocating independent tasks to parallel processors: an experimental study. *J. Par. Dist. Comp.*, 47:185–197, 1997.

[6] C. P. Kruskal and A. Weiss. Allocating independent subtasks on parallel processors. *IEEE Trans. Software Eng.*, 11:1001–1016, 1985.

[7] S. Lucco. A dynamic scheduling method for irregular parallel programs. In *Proc. SIGPLAN Conference on Programming Language Design and Implementation*, pages 200–211, 1992.

4.1.3 Deterministic Parallel Padded Sorting

Investigators: K.-W. Chong and Edgar A. Ramos

Given an input array of n real numbers, sorting with padding λ consists of writing those numbers in order in an array of size $(1+\lambda)n$, thus leaving λn entries *empty*. Only comparisons are allowed

between the numbers to be sorted. Although in the CRCW PRAM model there is a lower bound $\Omega(\log n/\log\log n)$ for *standard* sorting using a polynomial number of processors, it is possible to *pad-sort* faster. In particular, one is interested in studying the speed-up achievable when the *processor advantage* is k, *i.e.*, nk processors are available. Using randomization, the lower bound $\Omega(\log_k n)$ is matched by an algorithm by Hagerup and Raman [3]. They also gave a deterministic algorithm with running time $O(\log_k n(\log\log k)^5 \cdot 2^{C(\log^* n - \log^* k)})$ [4]. (This previous work by Hagerup and Raman was performed at the institute.) Goldberg and Zwick [2] reduced the exponent in the $\log\log k$ term to 3 by showing that approximate counting is possible in constant time with a polynomial number of processors in the CRCW PRAM model.

In [1], we describe a deterministic algorithm for the CRCW PRAM model that can pad-sort n numbers in time $O(\log_k n \log^*(\log_k n) + \log\log k)$ using nk processors with padding $\lambda = 1/\log^c(k + \log n)$. If $k = \log^\epsilon n$, for any constant $\epsilon > 0$, our algorithm provides a nearly optimal $o(\log n)$ time standard sorting algorithm: it sorts in time $O(\log n \log^* n/\log\log n)$ using $n\log^\epsilon n$ processors. This time is a factor $\log^* n$ greater than an optimal randomized algorithm by Rajasekaran and Reif [5]. Alternatively, the optimal time can be achieved with a processor advantage of $(\log n)^{\log^* n}$.

Our algorithm is a refinement of that in [4], which uses a natural divide-and-conquer approach. We essentially reduce the sorting problem to the problem of computing a *splitter* (a subset of the input that partitions the input into "nearly equal" intervals) and the corresponding intervals. At the base of the splitter computation, there is an algorithm that computes a splitter in time $O(1)$ using a polynomial number of processors; this is obtained by derandomizing a probabilistic argument through the method of limited independence. Then a slower but more processor-efficient algorithm is obtained by taking advantage of the properties of *samples*, a concept stronger than splitters. Finally, the general algorithm works by successive refinement using the previous algorithm. The algorithm relies essentially on fast approximate counting, and takes advantage of previous work on random sampling in the context of computational geometry.

References

- [1] K.-W. Chong and E. A. Ramos. Improved deterministic padded sorting. In *Proc. Annual European Symposium on Algorithms, Springer LNCS 1461*, pages 405–416, 1998.

[2] T. Goldberg and U. Zwick. Optimal deterministic approximate parallel prefix sums and their applications. In *Proc. 4th IEEE Israel Symp. on Theory of Comput. and Sys.*, pages 220–228, 1995.

[3] T. Hagerup and R. Raman. Waste makes haste: Tight bounds for loose parallel sorting. In *Proc. 33th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 628–637, 1992.

[4] T. Hagerup and R. Raman. Fast deterministic approximate and exact parallel sorting. In *Proc. ACM Sympos. on Parallel Algorithms and Architectures*, pages 346–355, 1993.

[5] S. Rajasekaran and J. Reif. Optimal and sublogarithmic time randomized parallel sorting algorithms. *SIAM J. Comput.*, 18:594–607, 1989.

4.1.4 Parallel Graph Algorithms

Single Source Shortest Paths (SSSP)

Investigators: Andreas Crauser, Kurt Mehlhorn, Ulrich Meyer, and Peter Sanders

Computing shortest paths is an important combinatorial optimization problem with numerous applications [1]. Sequentially, Dijkstra's algorithm with Fibonacci heaps computes the shortest path tree of an arbitrary vertex in a non-negative weighted digraph with n nodes and m edges

in time $O(n \log n + m)$. The algorithm gains its efficiency by maintaining the invariant that the tentative distance of the queued node with minimal priority always equals its final distance.

An easy parallel SSSP solution is to run Dijkstra with n phases and restrict parallelism to the relaxation of outgoing edges. However, inter-processor communication phases incur large startup costs that often dominate the total running time. Therefore, the n-phase algorithm only pays off for very dense graphs. On the other hand, there are approaches to solve the SSSP within just $O(\log^2 n)$ rounds, but they require an infeasible number of nearly $O(n^3)$ compute operations [4].

In our experiments we found by looking at the queued nodes during a run of Dijkstra that frequently many more nodes than the one with minimal priority had already reached their final distance; these nodes could be removed in parallel. We identified simple criteria to find such nodes [3]. Unfortunately, there are instances where our approach still needs n phases, but on several graph classes, including real world data, performance is quite good. For random graphs with random edge weights, only $O(\sqrt{n})$ phases are necessary for a simple criterion, $O(n^{1/3})$ phases for a more advanced method. We derive a PRAM algorithm that runs in $O(n^{1/3} \log n)$ time while performing just $O(n \log n + m)$ work with high probability (whp). The underlying idea can also be used to improve the performance of SSSP in external memory with parallel independent disks [2].

The above algorithms are appealing in the sense that the node removal criteria are self-adapting and do not need any adjustment to different edge weights or graph classes. However, they are often more restrictive than necessary. In [5] we investigate an algorithm, Δ-*stepping*, that always dequeues the nodes belonging to a complete interval of priorities. The approach may remove nodes with wrong distances, thus causing reinsertions. However, in the case of random graphs with random edge weights, uniformly chosen from $[0, 1]$, we showed that an interval size $\Delta = \Theta(n/m)$ keeps the number of reinsertions bounded by $O(n)$ whp; the number of phases is less than $O(\log^2 n)$ whp. Using a parallel data structure with buckets of breadth Δ instead of a priority queue, the total number of operations is even linear whp., $O(n + m)$, and we get a very efficient PRAM algorithm that achieves running time $O(\log^3 n)$ on those random graphs whp.

First implementations on an Intel Paragon revealed promising results: on 16 (64) processors, we obtained a speedup of up to 9 (30) for sparse, and 7.5 for dense random graphs compared to the sequential version of Δ-stepping. The latter in turn was up to three times faster than an optimized sequential implementation of Dijkstra's algorithm. Currently, we are developing improved Δ-stepping versions that remain efficient on more general graph classes.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows : Theory, algorithms and applications.* Prentice Hall, 1993.

- [2] A. Crauser, K. Mehlhorn, and U. Meyer. Kürzeste-Wege-Berechnung bei sehr großen Datenmengen. In *Promotion tut not: Innovationsmotor "Graduiertenkolleg"*, volume 21 of *Aachener Beiträge zur Informatik*, pages 113–132, Aachen, Germany, September 1997. Augustinus.

- [3] A. Crauser, K. Mehlhorn, U. Meyer, and P. Sanders. A parallelization of Dijkstra's shortest path algorithm. In *Proceedings of the 23rd International Symposium on the Mathematical Foundations of Computer Science (MFCS-98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 722–731, Brno, Czech Republic, August 1998. Springer.

[4] Y. Han, V. Pan, and J. Reif. Efficient parallel algorithms for computing all pair shortest paths in directed graphs. In *Proceedings of the 4th Annual Symposium on Parallel Algorithms and Architectures*, pages 353–362. ACM Press, 1992.

- [5] U. Meyer and P. Sanders. Δ-stepping: A parallel single source shortest path algorithm. In *Proceedings of the 6th Annual European Symposium on Algorithms (ESA-98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 393–404, Venice, Italy, August 1998. Springer.

List Ranking

Investigators: Jop F. Sibeyn

List ranking is an elementary problem: we have a set of linked lists, and all nodes should compute the indices of the final nodes of their lists and the distances thereto. Sequentially, this problem is trivial, but it is hard to parallelize. Algorithms that came to within a constant factor from optimality were known before, but we were not satisfied with their apparent sub-optimality. In practice these algorithms all perform very poorly. In an extensive study we have developed several new, more efficient algorithms [2, 1]. These algorithms have also been implemented on the Intel Paragon. In our case, the Paragon had 140 PUs and a mesh as the interconnection network. On a partition with P PUs, the best of our algorithms achieved speed-up of almost $P/3$. This means that if the sequential problem can be solved in t seconds, we can solve it with P PUs in $3 \cdot t/P$ seconds [3, 4]. The problem may not have so much importance by itself, but (at least in theory) it is an important subroutine, and our case analysis has clearly shown the possibilities and limitations of parallel computing for irregularly structured problems.

References

[1] J. F. Sibeyn. Better trade-offs for parallel list ranking. In *Proceedings of the 9th Symposium on Parallel Algorithms and Architectures (SPAA-97)*, pages 221–230, New-Port Beach, 1997. ACM Press, ACM Press.

- [2] J. F. Sibeyn. List ranking on meshes. *Acta Informatica*, 35(7):543–566, 1998.
- [3] J. F. Sibeyn, F. Guillaume, and T. Seidel. Practical parallel list ranking. In G. Bilardi, A. Ferreira, R. Lüling, and J. Rolim, editors, *Proceedings of the 4th Symposium on Solving Irregularly Structured Problems in Parallel (IRREGULAR-97)*, volume 1253 of *Lecture Notes in Computer Science*, pages 25–36, Paderborn, Germany, 1997. Springer.
- [4] J. F. Sibeyn, F. Guillaume, and T. Seidel. Practical parallel list ranking. *Journal of Parallel and Distributed Computing*, 56:156–180, 1999.

Undirected Connectivity and Minimum Spanning Trees

Investigator: Ka Wong Chong

Given a weighted undirected graph G with n vertices and m edges, the minimum spanning tree (MST) problem is to find a spanning tree (or forest) of G with minimized total edge weights. In the parallel context, the MST problem is closely related to the connected component (CC) problem, which is to find the connected components of an undirected graph. Techniques for solving the two problems in parallel are very similar. If concurrent write is allowed, it is relatively simple to solve both problems in $O(\log n)$ time using $n + m$ processors on the CRCW PRAM.

For the exclusive write models (*i.e.*, CREW and EREW PRAMs), $O(\log^2 n)$ time algorithms for the connected component and MST problems were developed two decades ago. For a while, it was believed that exclusive write models cannot overcome the $O(\log^2 n)$ time bound. The first breakthrough is due to Johnson and Metaxas; they devised $O(\log^{1.5} n)$ time algorithms for both problems. These results were further improved by Chong and Lam to $O(\log n \log\log n)$ time. If randomization is allowed, the time can be further improved to $O(\log n)$ [2].

Prior to our work, it had been open whether the connected component and MST problems could be solved deterministically in $O(\log n)$ time on the exclusive write models. Notice that $O(\log n)$ is optimal in view of the fact that these graph problems are at least as hard as computing the OR of n bits. It has been proven that the latter requires $\Omega(\log n)$ time on the CREW or EREW PRAM with no restriction on the number of processors.

We present a new parallel algorithm for the MST problem. It runs in $O(\log n)$ time using a linear number of processors on the EREW PRAM [1]. Our work resolves a long-standing open problem in the literature about whether the PRAM with the concurrent-write capability is more efficient for graph problems like connected components and minimum spanning trees than the PRAM without concurrent-write capability.

References

- [1] K.W. Chong, Y. Han, and T.W. Lam. On the parallel time complexity of undirected connectivity and minimum spanning trees. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 225–234, 1999.

[2] D. R. Karger. *Random sampling in Graph Optimization.* PhD thesis, Stanford University, 1995.

4.2 External Computing

4.2.1 Selection

Investigator: Jop Sibeyn

An algorithm with a slightly theoretical flavor has been designed for the selection problem [1]. We show that for selecting an element with prescribed rank, the median say, one needs to read and write all data only once. For a deterministic algorithm, this is probably the best that one can achieve (though randomly the problem can be solved faster).

References

- [1] J. F. Sibeyn. External selection. In *Proc. 16th Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 291–301. Springer, 1999.

4.2.2 Priority Queues

Investigators: Gerth Brodal, Andreas Crauser, Paolo Ferragina, Ulrich Meyer, and Peter Sanders

A *priority queue* (PQ) is a data structure that stores a set of items, each one consisting of a tuple that contains some *(satellite) information* plus a *priority* value (also called *key*) drawn from a totally ordered universe. A PQ supports the following operations on the processed set: `access_minimum` (returns the item in the set having minimum key), `delete_minimum` (returns and deletes the item in the set having the minimum key), `insert` (inserts a new item into the set) and possibly `decrease_key` (decreases the key of an item present in the set). PQs have numerous important applications: combinatorial optimization (*e.g.* Dijkstra's shortest path algorithm), time forward processing, job scheduling, event simulation and online sorting just to cite a few. Many PQ implementations currently exist for small data sets fitting into the *internal memory* of the computer, *e.g.*, k–ary heaps, Fibonacci heaps, and radix heaps. However, as soon as the PQ does not fit into the main memory any more (*e.g.*, for a large discrete event simulation), these internal algorithms become prohibitively slow.

Based on the work of Thorup for RAM-PQs, we designed PQ data structures based on merging sorted lists. The data structure consists of a hierarchy of i levels in secondary memory, each level consisting of $O(M/B)$ sorted lists (called also *slots*) of size $O(M^i/B^{i-1})$. The slots are merged upon level- or internal memory overflow. One variant achieves asymptotically optimal worst-case performance for the PQ operations `insert` and `delete_minimum` [5]. A simpler implementation, array heaps [4], is based on sorted arrays and achieves the same *amortized* complexity bounds for operations `insert` and `delete_minimum` of the previously best known results. We have further refined array heaps into *sequence heaps* [6], which require only $\frac{2I}{B}(\lceil \log_{\Omega(M/B)}(O(\frac{I}{M}))\rceil + \frac{1}{M}O(B + \log\frac{M}{B}))$ I/Os for any sequence of insertions and deletions containing I insertions. This is almost as fast as the special case of multi-way merge sort and at least three times faster than the previously mentioned algorithms.

We have also introduced a PQ for integer keys [4] that is based on internal memory radix heaps [1]. A practical comparison between the implementation of array heaps, radix heaps and other known external memory PQs (buffer trees [2], B-Trees [3]) shows their effectiveness. Sequence heaps turned out to be so efficient that they can also be used as a cache-efficient internal algorithm (Section 4.2.5).

References

[1] R. Ahuja, K. Mehlhorn, J.B. Orlin, and R.E. Tarjan. Faster algorithms for the shortest path problem. *Journal of the ACM*, 3(2):213–223, 1990.

[2] L. Arge. The buffer tree: A new technique for optimal I/O-algorithms. In *Lecture Notes in Computer Science*, volume 955, pages 334–345, 1995.

[3] R. Bayer and E. McCreight. Organization and maintenance of large ordered indizes. *Acta Informatica*, 1:173–189, 1972.

- [4] K. Brengel, A. Crauser, P. Ferragina, and U. Meyer. An experimental study of priority queues in external memory. *Submitted to Workshop on Algorithmic Engineering (WAE)*, 1999.
- [5] G. S. Brodal and J. Katajainen. Worst-case efficient external-memory priority queues. In *Lecture Notes in Computer Science*, volume 1432, pages 107–118, 1998.
- [6] P. Sanders. Fast priority queues for cached memory. In *ALENEX '99, Workshop on Algorithm Engineering and Experimentation*, LNCS. Springer, 1999.

4.2.3 Text-Indexing

Investigators: Andreas Crauser and Paolo Ferragina

In the information age, one of the fastest growing categories of databases ia textual databases like digital libraries, genome databases or book collections. The ultimate impact of these databases heavily depends on the ability to *store* and *search* efficiently the information contained in them. The continuous decline of storage cost has put the challenge of fast information retrieval in huge textual databases into the focus of interest. In order to achieve this goal, specialized indexing data structures and searching tools have been introduced. Their main idea is to build an *index* that allows the search for a given pattern string to focus on only a very small portion of the text collection. The improvement in the query-performance is paid for by additional space necessary to store the index. Most of the research in this field has therefore been directed to designing indexing data structures that offer a good trade-off between the query time and the space usage. Examples of such indexes are: suffix trees [9], suffix arrays [8], PAT-trees [7], Patricia trees [10], and Prefix B-trees [1]. They have been successfully used for fundamental string-matching problems as well

as for applications like text compression or analysis of genetic sequences [2]. These indexes are therefore the natural choice for performing fast complex searches without any restrictions. The most important complexity measures for evaluating their efficiency are: (i) the time and the extra space required to build the index, (ii) the time required to search for a string, and (iii) the space used to store the index.

We studied all these three issues in an external memory setting where the majority of the known indexing algorithms perform poorly. In [4, 3] we have investigated Point (i) above by addressing the efficient construction of two well-known data structures, namely suffix trees and suffix arrays, on very large text collections. Known algorithms for building suffix trees elicit many disk accesses when working on external storage devices. Our main contribution in [4] has been to propose a new algorithm for suffix tree construction in which we choreograph almost all disk accesses to be via the `sort` and `scan` primitives. The technique is general enough that we can apply it uniformly to several parallel and hierarchical-memory models, and thus obtain the first known optimal algorithms therein. Furthermore, since many solutions to string problems have the suffix-tree construction as their I/O-bottleneck, all these now have efficient implementations in external memory.

When the space occupancy is a primary concern, suffix trees are not very attractive and thus people usually turn their attention to the suffix-array data structure [8]. Suffix arrays have recently been the subject of experimental investigations in internal memory, external memory and distributed memory systems. But, to the best of our knowledge, no full range comparison exists among the known algorithms for building large suffix arrays. This has been the main goal of [3], where we have theoretically studied and experimentally analyzed six suffix-array construction algorithms. Some of them are the state-of-the-art in the practical setting, others are the most efficient theoretical ones, whereas three other algorithms are new proposals by us. As a result, [3] gives a precise hierarchy of suffix-array construction algorithms according to their experimented working-space vs. construction-time tradeoff, thus providing a wide spectrum of possible approaches for anyone who is interested in building large indexes.

Searching performance is also a primary issue to be considered when designing indexing data structures (see Point (ii) above). Classical tools for manipulating external texts – suffix trees, suffix arrays, prefix B-trees, inverted files – offer very good practical performance but are not optimal. In [5] we introduced a novel data structure, called a *string B-tree*, which allows one to circumvent two major difficulties that are encountered by the classical approaches to managing long strings. First, each long string is represented using a constant amount of space (independent of its length); second, it avoids the computational overhead usually incurred in comparing two long strings character-by-character. As a consequence, string B-trees make it possible to search arbitrary strings in a very large text archive with a negligible number of disk accesses, and they allow efficient maintainance of the archive under changes that are performed on it over time. There is preliminary evidence that the string B-tree performance is also good in practice, but further research will aim at validating these early experimental results.

The main ideas contained in the design of string B-trees have been exploited in [6] to devise a new indexing data structure suitable also for a *distributed* memory environment. Such a data structure makes it possible to reduce and balance the communication cost involved in a search for multiple pattern strings inside a textual archive that is distributed among p processors. In particular, the result turns out to be optimal in the case that the patterns are longer than p. This is fairly natural in practice. A typical case is the indexing in WEB servers, which usually consist of a few powerful commodity workstations connected by a high-speed local network. Experiments are now required to validate these theoretical results in the practical setting.

References

[1] R. Bayer and K. Unterauer. Prefix B–trees. *ACM Transactions on Database Systems*, 2(1), March 1977. Also published in/as: IBM Yorktwon, T.R.RJ1796, Jun.1976.

- [2] S. Burkhardt, A. Crauser, P. Ferragina, H.P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array. *International Conference on Computational Molecular Biology (RECOMB)*, pages 77–83, 1999.
- [3] A. Crauser and P. Ferragina. On constructing suffix arrays in external memory. *European Symposium on Algorithms (ESA)*, 1999.
- [4] M. Farach, P. Ferragina, and S. Muthukrishnan. Overcoming the memory bottleneck in suffix tree construction. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 174–183, 1998.
- [5] P. Ferragina and R. Grossi. The string B-tree: A new data structure for string search in external memory and its applications. *Journal of the ACM (To appear), Preliminary version in ACM STOC, 1995*, 1999.
- [6] P. Ferragina and F. Luccio. Multi-string search in BSP. In *SEQS: Compression and Complexity of Sequences 1997*. IEEE Press, 1998.

[7] G. H. Gonnet, R. A. Baeza-Yates, and T. Snider. New indices for text:PAT trees and PAT arrays. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval – Data Structures and Algorithms*, pages 66–82. Prentice-Hall, 1992.

[8] U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM J. Comput.*, 22(5):935–948, October 1993.

[9] Edward M. McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM*, 23(2):262–272, April 1976.

[10] D. R. Morrison. PATRICIA - practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM*, 15(4):514–534, October 1968.

4.2.4 Geometric Problems

Computer graphics and geographic information systems are nowadays rich sources of large-scale computational problems. Consequently, there is a need for external-memory techniques and data structures that can cope efficiently with the enormous amount of spatial data that have to be manipulated. This results in a rich source of problems for the field of computational geometry. Thus, we have considered the I/O-complexity of some geometric problems in the external memory model.

I/O-Efficient Dynamic Point Location in a Monotone Subdivision

Investigator: Gerth Stølting Brodal

In internal memory, Edelsbrunner *et al.* [4] proposed an optimal data structure for point location in monotone subdivisions with $O(N)$ space, $O(N)$ preprocessing time, and $O(\log N)$ query time. For arbitrary planar subdivisions, the preprocessing time is $O(N \log N)$. If the edges and vertices are allowed to be changed dynamically, two linear-space structures are known for general subdivisions: one by Cheng and Janardan [3] that answers queries in $O(\log^2 N)$ time and supports updates in $O(\log N)$ time; the other by Baumgarten *et al.* [2] that supports queries in worst-case $O(\log N \log\log N)$ time, insertions in amortized $O(\log N \log\log N)$ time, and deletions in amortized $O(\log^2 N)$ time.

In [1] we consider the problem of dynamically maintaining a monotone subdivision on disk, so that the number of I/Os used to perform a query or an update is minimized. The data structure presented in [1] uses $O(N/B)$ disk blocks to store a monotone subdivision, answers queries in $O(\log_B^2 N)$ I/Os in the worst-case, and inserts/deletes edges and vertices in $O(\log_B^2 N)$ I/Os amortized per edge/vertex, where B is the number of elements per disk block.

In order to answer the queries efficiently, we maintain a B-tree-like data structure supporting *split* and *merge* operations, where each node stores a pointer to its parent. Although merge and split operations on standard B-trees can be performed in $O(\log_B N)$ I/Os, updating the parent pointers requires $\Omega(B \log_B N)$ I/Os. We therefore introduce a new variant of B-trees called *level-balanced B-trees* in which parent pointers can be maintained efficiently. For $2 \leq b \leq B/2$, level-balanced B-trees use $O(N/B)$ blocks to store N elements, and support insert, delete, merge, and split operations in $O((1 + \frac{b}{B} \log_{M/B} \frac{N}{B}) \log_b N) = O(\log_B^2 N)$ I/Os amortized, where M is the number of elements fitting in internal memory, $M \geq 2B$.

References

- [1] P. Agarwal, L. Arge, G. S. Brodal, and J. Vitter. I/O-efficient dynamic point location in monotone subdivisions. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 11–20, 1999.

[2] H. Baumgarten, H. Jung, and K. Mehlhorn. Dynamic point location in general subdivisions. *Journal of Algorithms*, pages 342–380, 1994.

[3] S. W. Cheng and R. Janardan. New results on dynamic planar point location. *SIAM Journal of Computing*, 21:972–999, 1992.

[4] H. Edelsbrunner, L. J. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM Journal of Computing*, 15:317–340, 1986.

I/O-Efficient Randomized Incremental Construction

Investigators: A. Crauser, P. Ferragina, K. Mehlhorn, U. Meyer and E. A. Ramos

Since the *randomized incremental construction* approach [3, 6] has been successfully employed in the design of geometric algorithms in internal memory, a natural question was the possibility of extending it to obtain I/O-efficient algorithms.

A fundamental problem in computational geometry that also arises in many applications is the computation of the intersections of a set of line segments or, more generally, the corresponding trapezoidal decomposition (the diagram obtained by extending vertically each segment endpoint and each intersection point between segments until another segment is hit). Let N be the number of segments and let K be the number of intersection points. In internal memory, several deterministic and randomized algorithms are known that compute the trapezoidal decomposition or a variation of it [2, 3, 5]. In the external memory model, *optimality* means that the total number of I/Os required to compute the K intersections or the trapezoidal decomposition of S, is $\Theta(n \log_m n + k)$, where $n = N/B$, $m = M/B$ and $k = K/B$. Prior to our work, no I/O-optimal algorithm was known. The best known algorithms were due to Arge *et al.* [1]. One of their algorithms computes the intersections (but not the trapezoidal decomposition) in sub-optimal $O((n + k) \log_m n)$ I/Os, and another computes the trapezoidal decomposition induced by a set of *non-intersecting* segments in optimal $O(n \log_m n)$ I/Os.

In [4], we described how the randomized incremental construction approach can indeed be adapted to provide I/O-efficient algorithms. In this context, instead of adding the geometric objects one by one, they are added in groups of geometrically increasing size. Using this approach, we

showed that it is possible to compute the trapezoidal decomposition for a set of line segments using an optimal expected number of I/Os. The approach is sufficiently general to work for other problems: 3-d half-space intersections, 2-d and 3-d convex hulls, 2-d abstract Voronoi diagrams and batched planar point location. The resulting algorithms require an optimal expected number of disk accesses and are simpler than the ones previously known. The results extend to an external-memory model with multiple disks.

Additionally, under practical conditions on the parameters N, M, and B, these results can be notably simplified, thus providing practical algorithms that still achieve optimal expected bounds. We are currently working on the implementation of this simpler version of our algorithms in the framework of the `LEDA-SM` library in order to evaluate their practical efficiency on real-world problems.

References

[1] L. Arge, D. E. Vengroff, and J. S. Vitter. External-memory algorithms for processing line segments in geographic information systems. In *Proc. Annual European Symposium on Algorithms, Springer LNCS 979*, pages 295–310, 1995.

[2] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM*, 39:1–54, 1992.

[3] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

- [4] A. Crauser, P. Ferragina, K. Mehlhorn, U. Meyer, and E. A. Ramos. Randomized external-memory algorithms for some geometric problems. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 259–268, 1998.

[5] K. Mulmuley. A fast planar partition algorithm, I. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 580–589, 1988.

[6] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice Hall, 1993.

4.2.5 Modeling Caches

Investigator: Peter Sanders

Often, the RAM model with its uniform memory does not adequately model the performance characteristics of mainstream sequential computers because the access latencies between the first level cache and main memory can differ by up to two orders of magnitude. Therefore, we have investigated whether the external memory model and the algorithms developed for it might be a useful source of cache-efficient algorithms. Obviously, the analogy fits quite well since a cache of size M interacts with the main memory by reading and writing cache blocks of size B.

Our theoretical efforts have concentrated on the main conceptual difference between hardware caches and the external memory model: External memory algorithms have full control over the content of the internal memory whereas caches usually employ a fixed *a-way associative* strategy in which each block of the main memory is mapped to a unique *cache set* that can store up to a blocks (usually, $a \in \{1, 2, 4\}$). If a cache set overflows, the least recently used block is replaced by the new block. How do set-associative caches perform for the typical access pattern of some of the most successful external memory algorithms (*i.e.*, reading from many (up to $O(M/B)$) sources and writing to one (or a few) destinations (or vice versa))? The predominant theoretical model previously available — the independent reference model [2] — predicts very poor performance in

this situation regardless of the degree of associativity. We show that an a-way associative cache performs well if the number of streams is reduced to $O(M/B^{1+1/a})$ [4]. So, in particular for large cache lines, associativity has quite a big impact. This result even holds for worst case access patterns if the starting addresses of the sequences can be considered random.

A practically important difference between the cache-memory hierarchy and the memory-disk hierachy is that the speed gap is much smaller for the cache-memory case. Therefore, constant factors matter when we want to use external memory techniques for the design of cache-efficient algorithms. We started with simple experiments: Random permutations can be generated several times faster than with the classical RAM algorithm [1] if a distribution-based external algorithm is used [3] although this algorithm would be considered almost two times slower in the RAM model. Even a quite involved external list ranking algorithm [6] runs slightly faster than plain pointer chasing on recent machines. More effort was needed to adapt external memory priority queue algorithms. The constant factor improvements achieved by sequence heaps (Section 4.2.2) are really needed here. Now, a careful implementation is at least two times faster than an optimized implementation of binary heaps and 4-ary heaps for large inputs [5].

References

[1] D. E. Knuth. *The Art of Computer Programming — Seminumerical Algorithms*, volume 2. Addison Wesley, 2nd edition, 1981.

[2] G. Rao. Performance analysis of cache memories. *Journal of the ACM*, 25(3):378–395, 1978.

- [3] P. Sanders. Random permutations on distributed, external and hierarchical memory. *Information Processing Letters*, 67(6):305–310, 1998.
- [4] P. Sanders. Accessing multiple sequences through set associative caches. In *ICALP*, LNCS, 1999.
- [5] P. Sanders. Fast priority queues for cached memory. In *ALENEX '99, Workshop on Algorithm Engineering and Experimentation*, LNCS. Springer, 1999.
- [6] J. Sibeyn. From parallel to external list ranking. Technical Report MPI-I-97-1-021, Max-Planck Institut für Informatik, 1997.

4.2.6 Practical Software

Investigators: Andreas Crauser and Kurt Mehlhorn

In recent years, the algorithmic community has developed many I/O-efficient algorithms and data structures for many graph, string and geometric problems (see [6] for a survey), but implementations and experimental work are lacking behind.

In [1, 4] we propose *LEDA-SM* (LEDA secondary memory) as a platform for external memory computation. It extends LEDA [5] to external memory computation and is therefore directly connected to an efficient internal-memory library of data structures and algorithms. LEDA-SM is portable, easy to use, and efficient. It consists of:

- a kernel that gives an abstract view of external memory and provides a convenient infrastructure for implementing external memory algorithms and data structures. We view external memory as a collection of disks and each disk as a collection of blocks.
- a collection of external memory data structures. An external memory data structure offers an interface that is akin to the interface of the corresponding internal memory data structures (of LEDA), uses only a small amount of internal memory, and offers efficient access to external memory.

- algorithms operating on these data structures. This includes basic algorithms like sorting as well as matrix multiplication, text indexing and simple graph algorithms.
- a precise and readable specification for all data structures and algorithms. The specifications are short and abstract so as to hide all details of the implementation.

The external memory data structures and algorithms of LEDA-SM are based on the kernel; however, their use requires little knowledge of the kernel. LEDA-SM supports fast prototyping of external memory algorithms and therefore can be used to experimentally analyze new data structures and algorithms in an external storage setting. This was first done in [4] were all data structures and algorithms, available in LEDA-SM, were tested against their internal-memory counterpart of LEDA. In [3], we used the library to compare several different external-memory algorithms for suffix array construction (see also Section 4.2.3) and in [2] we used LEDA-SM to compare external memory priority queues (see also Section 4.2.2).

References

- [1] A.Crauser and K. Mehlhorn. LEDA-SM: Extending LEDA to secondary memory. *submitted to Workshop on Algorithmic Engineering (WAE)*, 1999.
- [2] K. Brengel, A. Crauser, P. Ferragina, and U. Meyer. An experimental study of priority queues in external memory. *Submitted to Workshop on Algorithmic Engineering (WAE)*, 1999.
- [3] A. Crauser and P. Ferragina. On constructing suffix arrays in external memory. *European Symposium on Algorithms (ESA)*, 1999.
- [4] A. Crauser, K. Mehlhorn, E. Althaus, K. Brengel, T. Buchheit, J. Keller, H. Krone, O. Lambert, R. Schulte, S. Thiel, M. Westphal, and R. Wirth. On the performance of LEDA-SM. Technical Report MPI-I-98-1-028, Max-Planck Institut für Informatik, 1998.
- [5] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999. The book is available at `www.mpi-sb.mpg.de/~mehlhorn`.

[6] J. S. Vitter. External memory algorithms and data structures. Survey Manuscript. `www.cs.duke.edu/~jsv/Papers/Catalog`.

5 Computational Geometry

Our work in this area combines the theoretical investigation of fundamental and application oriented questions with the actual implementation of algorithms and its corresponding theoretical support. We distinguish the following themes: theory and applications of geometric sampling, semidefinite optimization, external memory algorithms, applications, and implementations and their theoretical support.

We have continued work on the theory and applications of geometric sampling, both randomized and deterministic (Section 5.1). This has resulted in: the development of an I/O-efficient randomized incremental construction that extends the usual internal memory approach (see Section 4); some improved results on several fundamental problems in Computational Geometry – halfspace range reporting, ray shooting among hyperplanes and construction of levels in arrangements; the deterministic parallel solution of some *discrepancy* problems using a polynomial number of processors; and an improved algorithm for deterministic parallel *padded sorting* (Section 4).

In the field of semidefinite optimization (Section 5.2), our work includes work on its complexity when the dimension is fixed: extensions to semidefinite optimization of previous results on the linear-time solvability of linear programs, and on the polynomial solvability of integral programs. The latter work brings together techniques from Diophantine approximation and from convex and algorithmic real algebraic geometry.

Reflecting the general interest of the group in the area of external memory algorithms, we have developed algorithms and data structures for some geometric problems. This work is described in detail in Section 4. Our work in this area includes: an I/O-efficient randomized incremental approach (mentioned above), and an optimal data structure for dynamic point location in monotone subdivisions.

In regard to applications (Section 5.3), currently we are strongly interested in the problems of curve and surface reconstruction. We are pursuing theoretical aspects as well as actual implementations. Specifically, we are in the process of implementing several algorithms proposed in the literature and evaluating their performance. This includes algorithms for curve reconstruction that we have proposed. Another topic motivated in applications is robot motion planning. There is recently some activity due to the recent arrival of a member. Some work on the problem of motion planning for multiple tethered robots has been completed.

The work on implementation of geometric algorithms is described in detail in Section 8. Here (Section 5.4), for completeness, we give a short summary and then describe some work regarding theoretical support: structural floating-point filtering, and verification of Voronoi diagrams of lines segments.

5.1 Theory and Applications of Geometric Sampling

Halfspace Range Reporting, Ray Shooting and k-Level Construction

Investigator: Edgar A. Ramos

In [7], we consider some fundamental problems in computational geometry. Though they have been "essentially" solved in the past, we make progress in reducing the already narrow gap with respect to the trivial or conjectured lower bounds. The main tool throughout is geometric sampling.

Halfspace Range Reporting. Let P be a set of n points in d-space R^d. The problem is to preprocess P so that for a given query halfspace γ, the points $P \cap \gamma$ can be reported quickly. The important parameters are the preprocessing time, the storage space and the query answering time. The

latter depends also on the number $k = |P \cap \gamma|$ of points reported. A great amount of work has been performed on this problem. Recently, Chan [2] described a data structure for 3-space with expected preprocessing time $O(n \log n)$, worst-case space $O(n \log n)$ and expected query time $O(\log n + k)$. The case $d \geq 4$ had been solved a while ago almost "optimally" by Matoušek [4] with a deterministic data structure using preprocessing time $O(n \log n)$, space $O(n \log \log n)$ and query time $O(n^{1-1/\lfloor d/2 \rfloor} \log^c n + k)$, where c is a constant. We have improved on Chan's data structure by reducing the space to $O(n \log \log n)$ and by achieving the same query time, but worst-case, while maintaining the same optimal expected preprocessing time (furthermore, we somewhat simplify Chan's original data structure). For d even, we also obtain a reduction to storage $O(n)$ in Matoušek's data structure.

Ray Shooting. Let H be a set of n hyperplanes in R^d. A first problem, *ray shooting in a convex polytope,* is to preprocess H so that for a given query ray ρ with starting point p in the upper cell of H, the first hyperplane in H hit by ρ can be determined quickly. In particular, one is interested in achieving the smallest storage space that achieves a query time $O(\log n)$. There is the conjecture that query time $Q(n)$ and storage space $S(n)$ satisfy the relation $Q(n) \cdot S(n)^{1/\lfloor d/2 \rfloor} = \Omega(n)$. Matoušek and Schwarzkopf [6] achieve $Q(n) \cdot S(n)^{1/\lfloor d/2 \rfloor} = O(n \log^\delta n)$ (for small δ) using randomization in the construction. We described how to obtain deterministically query time $O(\log n)$ with preprocessing time and storage $O((n/\log n)^{\lfloor d/2 \rfloor} (\log \log n)^c)$, which corresponds to $Q(n) \cdot S(n)^{1/\lfloor d/2 \rfloor} = O(n(\log \log n)^c)$. With a somewhat larger preprocessing time, $O((n/\log n)^{\lfloor d/2 \rfloor} (\log n)^c)$, it is possible to achieve storage $O((n/\log n)^{\lfloor d/2 \rfloor} 2^{c \log^* n})$. A second problem, *ray shooting among hyperplanes,* is to construct a data structure so that for a given query ray ρ, the first hyperplane in H hit by ρ can be determined quickly. The particular case in which the ray is vertical was solved optimally with preprocessing time and storage $O((n/\log n)^d)$, and query time $O(\log n)$ [3, 5]; but for the general case, the best solution known has a query time $O(\log^2 n)$ [6]. We show that a query time $O(\log n)$ can be achieved while using storage $O(n^d/\log^{\lfloor d/2 \rfloor} n)$.

Construction of k-levels in 3-space. Let H be a set of n planes in R^3. The problem is to construct the *k-level* of H, namely, the set of those faces in the arrangement of H on the boundary of the region consisting of points with at most k planes below. We consider the particular case in which the planes in H are *dual* to points in convex position (by a standard geometric transformation, the 2-dimensional k-order Voronoi diagram of n sites corresponds to the k-level in one such arrangement). For this case, a tight bound $\Theta(nk)$ on the size is known, and it leads to a lower bound $\Omega(n \log n + nk)$ for constructing a k-level. Despite much work, an algorithm whose running time matches that lower bound has not been found. Recent breakthroughs were by Agarwal *et al.* [1] and by Chan [2]. The first paper gives an algorithm with expected running time $O(n \log^3 n + nk \log n)$, and the second reduces it to $O(n \log n + nk \log k)$. We have further reduced this time to $O(n \log n + nk 2^{\log^* k})$. This gives some evidence for the existence of an algorithm that matches the lower bound.

References

[1] P. K. Agarwal, J. Matoušek M. de Berg, and O. Schwarzkopf. Constructing levels in arrangements and higher order voronoi diagrams. *SIAM J. Comput.*, 27:654–667, 1998.

[2] T. Chan. Random sampling, halfspace range reporting and construction of ($\leq k$)-levels in three dimensions. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, 1988.

[3] B. Chazelle and J. Friedman. Point location among hyperplanes and unidirectional ray-shooting. *Comput. Geom.*, 4:53–62, 1994.

[4] J. Matoušek. Reporting points in halfspaces. *Comput. Geom.*, 2:169–186, 1992.

[5] J. Matoušek. On vertical ray shooting in arrangements. *Comput. Geom.*, 3:279–285, 1993.

[6] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10:215–232, 1993.

- [7] E. A. Ramos. On range reporting, ray shooting and k-level construction. In *Proc. 15th Annu. Sympos. Comput. Geom.*, 1999.

Solving Some Discrepancy Problems in NC

Investigators: Sanjeev Mahajan, Edgar A. Ramos and K. V. Subrahmanyam

Discrepancy is an important concept in combinatorics and theoretical computer science. It attempts to capture the idea of a *good sample* from a set. The simplest example, the *set discrepancy problem*, considers a *set system* $(X, \mathcal{S})$ where X is a ground set and $\mathcal{S} \subseteq 2^X$ is a family of subsets of X, and asks for a subset $R \subseteq X$ such that for each $S \in \mathcal{S}$ the difference $||R \cap S| - |\overline{R} \cap S||$, called the *discrepancy*, is small. Using Chernoff-Hoeffding bounds, it is found that a random sample $R \subseteq X$, with each $x \in X$ chosen for R independently with probability $1/2$, is with nonzero probability a low discrepancy set: for each $S \in \mathcal{S}$, $||R \cap S| - |\overline{R} \cap S|| = O(\sqrt{|S| \log |\mathcal{S}|})$. Sequentially, the method of *conditional probabilities* has been used to obtain an efficient deterministic algorithm that computes such a sample R [8]. In parallel, several approaches have been used (k-wise independence combined with the method of conditional probabilities and relaxed to biased spaces [1, 5, 6, 2]), but these efforts have resulted only in discrepancies $O(\sqrt{|S|^{1+\epsilon} \log |\mathcal{S}|})$. A similar situation happens in geometric sampling for set systems with constant VC-dimension. There, a sample of size $O(r^2 \log r)$ is known to be a $(1/r)$-*approximation* and it can be computed by an efficient deterministic sequential algorithm (via derandomization), but only size $O(r^{2+\epsilon})$ was achieved in parallel previously.

In [4], we describe NC algorithms (the algorithms run in $O(\log^2 n)$ time using a polynomial number of processors in the EREW PRAM model) that achieve the probabilistic bounds for discrepancy within a multiplicative factor $1+o(1)$. The technique we use is to model random sampling by *randomized finite automata* (finite automata in which transitions from a state to its immediate successor occurs with a certain probability), abbreviated RFA, and then *fool* these automata with a probability distribution of polynomial size support. The approach is not new; in fact, Karger and Koller [3] show how to fool such automata via the *lattice approximation* problem, using a solution for that problem developed in [5]. However, they did not realize that the lattice approximation problem can itself be modeled by RFAs and, as a result this and other discrepancy-like problems can be solved in parallel, nearly achieving the probabilistic bounds. We also describe how the work of Nisan [7] on fooling RFAs via pseudo-random generators also fits the same general approach. Although limited, the framework includes the lattice approximation problem, the discrepancy problem, and sampling problems in computational geometry (including computing a $(1/r)$-approximation of size $O(r^2 \log r)$). It also results in some improvements for parallel algorithms for approximate linear programming and graph edge coloring.

References

[1] B. Berger and J. Rompel. Simulating $(\log^c n)$-wise independence in NC. *Journal ACM*, 38:1026–1046, 1991.

[2] S. Chari, P. Rohatgi, and A. Srinivasan. Improved algorithms via approximations of probability distributions. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 584–592, 1994.

[3] D. R. Karger and D. Koller. (de)randomized constructions of small sample spaces in NC. In *Proc. 35th Annu. IEEE Sympos. Foundations Comput. Sci.*, pages 252–263, 1994.

- [4] S. Mahajan, E. A. Ramos, and K. V. Subrahmanyam. Solving some discrepancy problems in NC. In *Proc. 17th. Conf. Found. Soft. Techn. and Theoret. Comp. Sci., Springer LNCS 1346*, pages 22–36, 1997. To appear in *Algorithmica*.

[5] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. *J. Comput. Syst. Sci.*, 49:478–516, 1994.

[6] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22:838–856, 1993.

[7] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12:449–461, 1992.

[8] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *J. Comput. Syst. Sci.*, 37:130–143, 1988.

5.2 Semidefinite Programming

Recently, there has been substantial interest in semidefinite programming (SDP). Semidefinite programming can be regarded as an extension of linear programming in which the positive orthant is replaced by the cone of symmetric positive semidefinite matrices. Many convex optimization problems, *e.g.,* linear and convex quadratically constrained quadratic programs, maximum eigenvalue and matrix norm minimization, and also the computation of extremal ellipsoids for polyhedral sets, can be cast as SDP [12]. Applications of semidefinite programming include system and control theory, statistics, and combinatorial optimization.

It is well known that approximately solving semidefinite programs with explicitly given bounds on the size of an optimal solution can be accomplished in polynomial time by interior-point methods [1, 12]. However, the complexity of the general SDP problem remains an open fundamental problem of mathematical programming. In fact, it is not even known whether for the standard bit model of computation, the problem of testing the feasibility of a given semidefinite program belongs to the complexity class NP. Since the complexity status of the general SDP problem seems to be a very difficult question, it is natural to ask what other known complexity results for linear programming can be extended to semidefinite programming.

Semidefinite Optimization in Linear Time When the Dimension is Fixed

Investigator: Lorant Porkolab

Megiddo's result [11] on the linear-time solvability of linear programs in fixed dimension is a classical complexity result in the theory of linear programming. Recently we have extended [9] this result to the general semidefinite optimization problem: Compute the infimum of a linear objective function of an $n \times n$ symmetric positive semidefinite matrix satisfying a given system of m linear constraints; if the infimum is attained, find the least-norm optimal solution. We show the following result: *The general semidefinite optimization problem can be solved in* $mn^{O(n^2)}$ *arithmetic operations in the real number model of computation. Moreover, if the input coefficient are integers of binary length at most* l*, the required accuracy of arithmetic operations does not exceed* $ln^{O(n^2)}$ *bits.* These bounds, stated for the real number model of computation, include our earlier results [13], where we proved that in the standard bit model of computation the feasibility of general semidefinite programs with integral input coefficients of binary size at most l can be checked $mn^{O(n^2)}$ arithmetic operations over $ln^{O(n^2)}$-bit numbers. Another corollary of our bounds are nearly tight estimates on the algebraic degrees and logarithmic heights of the infimum and coordinates of the least-norm optimal solution.

To obtain the above results first we define parametric solutions to formulate semidefinite optimization as an LP-type problem – a notion introduced by Sharir and Welzl [15] – then we apply Chazelle and Matoušek's [4] derandomized variant of Clarkson's algorithm [5] along with the currently best known decision methods for the first-order theory of the reals due to Renegar [14], and Basu, Pollack and Roy [3].

Integer Optimization on Convex Semi-algebraic Sets

Investigator: Lorant Porkolab

Lenstra's result on the polynomial-time solvability of integer linear programming in fixed dimension is another well known complexity result in optimization. The question whether this can be extended to integer semidefinite programming motivated our work in [7], where we consider the problem of computing an integral point in an arbitrary convex semi-algebraic set (*i.e.,* the solution set of an arbitrary first-order algebraic formula with free and quantified variables). By applying a quantitative version of Kronecker's theorem on simultaneous Diophantine approximation, and some recent bounds [3] on the combinatorial and algebraic complexity of quantifier elimination methods for the first order theory of the reals, we first obtain an upper bound on the minimum binary size of an integral point contained in a given convex semi-algebraic set. Then we show that this bound implies the following generalization of the celebrated result of Lenstra [10]: *For each fixed n, there exists a polynomial-time algorithm that, given a convex semi-algebraic set defined by a first-order formula with n free and quantified variables, checks whether the input set contains an integral point, and if so, computes one.*

In addition to linear integer programming, this readily implies the polynomial-time solvability of systems of convex and quasi-convex polynomial inequalities with any fixed number of integer variables [6, 2]. It should be mentioned, however, that the above complexity result is more robust – it only uses the convexity of the solution set and does not require that each algebraic constraint be quasi-convex. In particular, it leads to the following corollary for integer semidefinite programming: *For each fixed n, there is a polynomial-time algorithm which finds an $n \times n$ integral symmetric positive semidefinite matrix satisfying a given system of linear inequalities, or decides that no such matrix exists.* This corollary also holds for systems of strict and/or nonstrict linear inequalities in positive definite and/or semidefinite matrices with integer and/or real variables, *i.e.,* for mixed SDP.

The previously mentioned results of [7] can also be extended to the optimization versions of the problems [8].

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] B. Bank, T. Krick, R Mandel, and P. Solerno. A geometrical bound for integer programming with polynomial constraints. In *Proceedings of the 8th International Conference on Fundamentals of Computation Theory*, LNCS 529, pages 121–125. Springer, 1991.

[3] S. Basu, R. Pollack, and M.-R. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43:1002–1045, 1996.

[4] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *Journal of Algorithms*, 21(3):579–597, 1996.

[5] K.L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM*, 42:488–499, 1995.

[6] L. Khachiyan. Convexity and complexity in polynomial programming. In *Proceedings of the International Congress of Mathematicians*, pages 1569–1577, Warszawa, August 16-24 1983.

- [7] L. Khachiyan and L. Porkolab. Computing integral points in convex semi-algebraic sets. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 162–171, 1997.
- [8] L. Khachiyan and L. Porkolab. Integer optimization on convex semi-algebraic sets. *Discrete and Computational Geometry*, To appear, 1999.
- [9] L. Khachiyan and L. Porkolab. Semidefinite optimization in linear time when the dimension is fixed. Manuscript, presented at the DIMACS Workshop on Semidefinite Programming, Princeton, NJ, January 1999.

[10] H.W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[11] N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31:114–127, 1984.

[12] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods for Convex Programming: Theory and Applications*. SIAM, Philadelphia, 1994.

[13] L. Porkolab and L. Khachiyan. On the complexity of semidefinite programming. *Journal of Global Optimization*, 10(4):351–365, 1997.

[14] J. Renegar. On the computational complexity and geometry of the first order theory of the reals. Part I: Introduction; preliminaries; the geometry of semi-algebraic sets; the decision problem for the existential theory of the reals. *Journal of Symbolic Computation*, 13:255–299, 1992.

[15] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, LNCS 577, pages 596–579. Springer, 1992.

5.3 Applications

Curve Reconstruction

Investigators: Tamal K. Dey, Kurt Mehlhorn and Edgar A. Ramos

Given a set of points in the plane that closely sample a curve, the task is to reconstruct (an approximation to) the original curve. This is a task easily performed by humans and its reproduction by computers is an important problem in computer vision, image processing and pattern recognition. Thus, the problem has drawn a lot of attention in the last three decades. However, only recently have solutions with a performance guarantee been proposed. If the curve is closed and uniformly sampled, a number of methods are known to work ranging over minimum spanning tree [5], α-shapes [2], β-skeleton [6], and r-regular shapes [1]. The case of non-uniformly sampled closed curves was first treated successfully by Amenta, Bern and Eppstein [7].

The problem is made precise as follows. A *curve* Γ is a collection of isolated points and single smooth curves that are pairwise disjoint. A point set $P \subseteq \Gamma$ is an *ϵ-sample* from Γ if for each point $p \in \Gamma$, its distance to its closest point in P is at most a fraction ϵ of the least distance from p to the medial axis of Γ (the closure of all center points of balls *touching* Γ in two or more points). The *correct reconstruction* of Γ from P is the graph with vertex set P such that for each $x, y \in P$, x and y are adjacent iff x and y are adjacent on Γ.

Amenta, Bern and Eppstein presented an algorithm CRUST that, given an ϵ-sample P from a closed curve for some $\epsilon < 0.252$, computes the correct reconstruction. Later Dey and Kumar [3] gave an algorithm NN-CRUST that works for $\epsilon < 1/3$. If P is not an ϵ-sample from a closed curve, no claim is made about the output of either algorithm. In [4], we describe an algorithm CONSERVATIVE-CRUST that, given a point set P and a non-negative real parameter ρ, constructs a graph G and a curve Γ. The graph G is a collection of open and closed chains with vertices in P satisfying the following properties:

1. *If P is a sufficiently dense sample from a curve, then G captures all edges in its correct reconstruction:* For $\rho < 1/2$, if P is a $(\rho/8)$-sample from a curve Γ', then G contains the correct reconstruction of Γ'.

2. *Γ justifies G:* For $\rho < 1/8$, there is a constant $c_0 \leq 13.35$, such that P is a $(c_0\rho)$-sample from Γ and G is the correct reconstruction of Γ.

3. *The algorithm can be implemented so that its running time is $O(n \log n)$.*

Unlike CONSERVATIVE-CRUST, the former two algorithms include edges in the reconstruction which one might call "unjustified".

References

[1] D. Attali. r-regular shape reconstruction from unorganized points. In *Proc. 13th Ann. Sympos. Comput. Geom.*, pages 248–253, 1997.

[2] F. Bernardini and C. L. Bajaj. Sampling and reconstructing manifolds using α-shapes. In *Proc. 9th Canadian Conf. Comput. Geom.*, pages 193–198, 1997.

- [3] T. K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *Proc. 10th. SIAM Sympos. Discr. Algorithms*, pages 893–894, 1999.

- [4] T. K. Dey, K. Mehlhorn, and E. A. Ramos. Curve reconstruction: Connecting dots with good reason. In *Proc. 15th Ann. Sympos. Comput. Geom.*, 1999.

[5] L. H. Figueiredo and J. M. Gomes. Computational morphology of curves. *Visual Computer*, 11:105–112, 1994.

[6] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In G. Toussaint, editor, *Computational Geometry*, pages 217–248. North Holland, 1985.

[7] M. Bern N. Amenta and D. Eppstein. The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60:125–135, 1997.

Multiple-Robot Motion Planning

Investigator: Susan Hert

Motion planning problems are generally stated in the following manner: given a starting location and a target location for a robot, construct a path between the two that avoids collisions with obstacles in the environment. When multiple robots are moving in the same environment, one must construct not only a path for each robot but, in order to avoid collisions among the robots, one must also provide a time profile of the robots' motion. Computing the path and the time profile simultaneously is an inherently difficult problem [1, 8]. Thus, many researchers (*e.g.*, [4, 6]) have addressed multiple-robot motion planning as a three-step process: First, choose an ordering for the robots; second, plan the paths for the robots assuming they move sequentially in the chosen

order; finally, compute the trajectories of the robots along these paths that assure the robots will not interfere with each other when moving simultaneously. Though this approach often makes it impossible to discover a solution even though one exists, in the problem we consider this loss of completeness is avoided under some minor assumptions about the starting and target locations of the robots.

In [5], we have considered the problem of motion planning for multiple robots in three dimensions. This problem is motivated by an application in underwater robotics, where the robots move in their three-dimensional environment to explore, inspect, or recover objects on the ocean floor. Each robot is attached by a cable (or *tether*) to a ship or platform on the surface of the ocean. The cables are used to transmit power or data to or from the robots. When planning motion for these robots it is important to make sure that the tethers do not become intertwined or *tangled*.

With the goal in mind of avoiding tangles altogether, we model the cables as straight line segments in R^3. Under this model we are able to analyze the potential interactions among the robots' tethers when they move along straight lines to their targets to derive an ordering of the robots that results in a maximum number of robots being able to move in straight lines. Our model also gives rise to a set of virtual three-dimensional obstacles, which, if avoided, will prevent the tethers from becoming tangled when the robots move sequentially in a given order. Canny and Reif [2] have shown that finding the shortest path between two points in three dimensions in the presence of obstacles is computationally intractable. This has led to the development of polynomial-time algorithms that produce approximately optimal paths (*e.g.*, [3, 7]). Through the use of such algorithms, we are able to compute paths for the robots that are nearly optimal if they move sequentially. We have shown that finding an optimal simultaneous-motion plan for the robots along a given set of paths is also computationally intractable, and have developed a polynomial-time algorithm for analyzing the potential interactions caused by motion along the paths and producing trajectories along the paths that are guaranteed to result in no collisions or tangles of the robots' tethers.

References

[1] J. Canny. *The Complexity of Robot Motion Planning.* ACM Doctoral Dissertation Awards. MIT Press, 1988.

[2] J. Canny and J. H. Reif. New lower bound techniques for robot motion planning problems. In *Proceedings 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 49–60, 1987.

[3] K. L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings 19th Annual ACM Symposium on Theory of Computing*, pages 56–65, 1987.

[4] M. Erdmann and T. Lozano-Pérez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.

- [5] S. Hert and V. Lumelsky. Motion planning in R^3 for multiple tethered robots. *IEEE Transactions on Robotics and Automation*, To appear.

[6] K. Kant and S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *International Journal of Robotics Research*, 5(3):72–89, 1986.

[7] C. H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Information Processing Letters*, 20:259–263, 1985.

[8] P. Spirakis and C. K. Yap. Strong NP-hardness of moving many discs. *Information Processing Letters*, 19(1):55–59, July 1984.

5.4 Implementation of Geometric Algorithms

5.4.1 Summary of Implementation Work

Investigators: Christoph Burnikel, Stefan Funke, Susan Hert, Stefan Schirra, Michael Seel, Mark Ziegelmann

It is now widely recognized also in the computational geometry community that making implementations of the developed algorithms available is an important step for the technology transfer from academia to industry. The best way to provide reusable implementations is to provide it in a software library. Our implementation work on geometric algorithms is made available in the software libraries CGAL and LEDA. Both libraries provide geometry kernels containing geometric primitives and a collection of algorithms and data structures, which subsumes by now most of the basic functionality described in the textbooks on computational geometry. Our work on the implementation of geometric algorithms is presented in Section 8.

5.4.2 Theoretical Support

Structural Filtering

Investigators: Stefan Funke, Kurt Mehlhorn, Stefan Näher

Floating-point filters have proved to be very efficient both in practice [2] and in theory [1] to speed up the exact evaluation of geometric predicates. If the input data is in general position, they allow exact evaluation of predicates with only a moderate overhead compared to pure floating-point arithmetic evaluation (the best non-static filters have an overhead of a factor of 2). Still, this overhead exists. Apart from that, degenerate or nearly degenerate tests pose a problem for these filters — they mostly fail and hence expensive arbitrary precision arithmetic has to be used. In practice, these "difficult" tests – though they occur only very rarely – affect the overall running time considerably.

Now the question is how to speed up the implementations even further. First, it would be nice to (almost) get rid of the overhead of floating-point filters in the "easy" cases. Second, we may want to decrease the number of "difficult" tests that have to be decided exactly.

By using only floating-point arithmetic for a majority of the predicate evaluations (and hence allowing some of the predicates to be decided incorrectly), we can save the overhead of the filter computations and possibly some arbitrary precision evaluations. Of course, we have to take care that the correct final result is still computed. We call this technique "structural filtering" where only the final result is guaranteed to be correct in contrast to predicate filtering where each predicate is guaranteed to be exactly evaluated. We have investigated the following problems:

- sorting: What happens if we allow comparisons to err ? How can we "repair" the result ?
- search structures: Can we use inexact comparisons on search structures and still get a correct result ?

Sorting. Assume we want to sort elements $x_1, ..., x_n$. Our comparison function used for sorting may err in a comparison of i, j, if $|rank(x_i) - rank(x_j)| < k$. As a measure of the quality of the outcome, we count the number of inversions.

We were able to prove that quicksort is optimal in this model up to a constant factor, and that mergesort is suboptimal. Note that an (almost) sorted sequence containing I inversions can be

sorted using finger search trees with $O(n \cdot \log(2 + I/n))$ (exact) comparisons or using insertion sort with $O(n + I)$ comparisons.

Practical experiments with quicksort (and insertion sort after each recursion) show a performance gain of roughly 20 percent, which is due to saving the computation of the error bounds for most predicates.

Searching. If the search structure is a directed acyclic graph, one can use inexact comparisons without the risk of looping. For example, if we have constructed a binary search tree for a set of n elements, and we assume the same "inexact" comparison function as for sorting, one can show that the leaf we end up with is at most k steps away from the "correct" one, and can be reached by walking along the leaves.

In principle, this works for all search structures that are directed acyclic graphs, where we can get from the possibly incorrect sink to the correct one afterwards. If the search structure is not a tree, it is also self-correcting in a sense that even if a wrong decision has been taken on the way, the correct sink may be reached.

Practical experiments for a search structure within the randomized incremental algorithm for computing a Delaunay triangulation showed a speedup for the point location of around 25 percent.

References

[1] Olivier Devillers and Franco Preparata. A probabilistic analysis of the power of arithmetic filters. *Discrete and Computational Geometry*, 20:523–547, 1998.

- [2] S. Schirra. A case study on the cost of geometric computing. In *Proc. of ALENEX'99*, 1999. To appear.

Verification of Voronoi Diagrams of Line Segments

Investigators: Christoph Burnikel, Kurt Mehlhorn and Michael Seel

Algorithms for computing the Euclidean Voronoi diagram of line segments are hard to implement. One reason is the numerical complexity of the problem [1]. Another is the geometric complexity of this particular Voronoi diagram that is due to an abundance of geometric cases that an implementation has to take into account. How can we be sure that the output of a program is the correct Voronoi diagram of line segments? Is there a *simple* and efficient procedure that rigorously proves or disproves the correctness of the returned graph G for an arbitrary set of input sites? In [2], we answer this question positively, presenting a new program checker for the Voronoi diagram of line segments in the spirit of [3].

The critical part of the checker is to show the planarity of the computed embedding. The key concepts used in our checker are the winding number and the orientation of curves. Using these two concepts, we show that it is enough to check for only *one* particular face of the graph G that it is a simple curve, namely its unbounded face. Our techniques can be generalized to every type of Voronoi diagram where the faces are convex or star-shaped.

References

[1] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections.* PhD thesis, Universität des Saarlandes, 1996.

- [2] C. Burnikel, K. Mehlhorn, and M. Seel. *A simple way to recognize a correct Voronoi diagram of line segments.* Research Report MPI-I-1999-1-003, Max-Planck-Insitut für Informatik, 1999.

[3] K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, R. Seidel, M. Seel, and C. Uhrig. Checking geometric programs or verification of geometric structures. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 159–165, 1996.

6 Computational Molecular Biology & Chemistry

We continued our work on sequence alignment problems and on the protein docking problem. In 1997 we started to design and implement a C++ library for Computational Molecular Biology. Since Summer 1998 our Computational Molecular Biology group has been working on an interdisciplinary project called GELENA (GEne transfer systems based on LEctin-modified NAnoparticles). We completed our research on efficient parallel algorithms for molecular dynamics simulation of synthetic polymers.

Since sequence alignment is of central importance in the interpretation of protein and DNA sequences, dozens of algorithms for the different alignment problems have been published in the last decades. Almost all of the published non-heuristic algorithms for these problems use dynamic programming. Since the time and space complexity of dynamic programming algorithms grows exponentially with the number of sequences, such approaches can solve only rather small problem instances to optimality, even if elaborate bounding procedures are applied. We tested whether new techniques from the area of combinatorial optimization can be successfully applied to sequence alignment problems. We studied the so-called Generalized Maximum Trace (GMT) problem and the Structural Maximum Trace (SMT) problem. We found integer linear programming (ILP) formulations for both alignment problems. An intensive study of the solution polyhedrons led us to branch-and-cut algorithms that are able to solve problem instances that cannot be handled by dynamic programming approaches. For example, the branch-and-cut-algorithm for the SMT problem is able to align two RNA sequences of length 1400. The best dynamic programming approach has a time and space complexity of $O(n^4)$ where n is the length of the longest RNA sequence. It can only solve problem instances of length up to 300. Furthermore, Knut Reinert presents in his PhD thesis an integer linear programming formulation of the "standard" alignment problem with arbitrary gap costs. In summary, we have developed a new approach that can be applied to many alignment problems. Although our current branch-and-cut algorithms already seem to be superior to dynamic programming approaches, there is still a lot of space to improve our algorithms by studying the solution polyhedrons and by identifying new facet-defining inequalities.

An Expressed Sequence Tag (EST) is a DNA fragment of length around 500 that stems from an expressed gene. The EST clustering problem can be defined as follows: Given a database of ESTs, determine for each EST all other ESTs in the database that stem from the same gene. These clusters can be computed by determining all pairs of overlapping ESTs in the database and storing this overlap information in a graph in which the nodes represent the ESTs and the edges represent the overlaps between two ESTs. Each connected component of this overlap graph represents an EST cluster. Because of errors in the EST data, algorithms for approximate matching have to be used to find the overlaps. The most commonly used approximate matching programs (BLAST and FASTA) were not designed for multiple high similarity searches. For example, we estimate that the clustering of the NCBI UNIGENE human EST database which contains roughly 725,000 ESTs would take six months on an average workstation with one processor using BLAST. In close cooperation with the German Cancer Research Center, we developed and implemented a specialized database search algorithm (QUASAR). It is designed and optimized for high similarity search. Typical applications of our new algorithm are EST database clustering and sequence assembly. The algorithm uses a suffix array as an index data structure. Thus, in contrast to BLAST and FASTA, it does not scan the whole database. The combination of a suffix array with block addressing schemes and other well known techniques leads to a database search engine that achieves a speedup factor of about 30 against the NCBI BLAST 2 program.

In 1996 we started to design and implement a C++ library, called BALL (Biochemical ALgo-

rithms Library), for rapid software prototyping in the area of molecular modeling and simulation. Besides the library kernel, the current version of BALL provides modules for Molecular Mechanics and for solvation and electrostatic energy calculations based on a Poisson-Boltzmann solver. The visualization component BALLVIEW provides portable visualization of the kernel data structures and general geometric primitives. In a first test with BALL we were able to re-implement a program in one day, the first implementation of which, in the course of a Master's thesis, took half a year. First practical experiences in our protein docking project (see below) were also very successful, because we were able to implement complex energetic calculations in a few weeks.

The goal of protein docking research is the development of algorithms that enable the user to predict reactions between proteins and to compute the three-dimensional structure of the resulting protein complexes. Protein docking algorithms can be used to screen protein databases for possible inhibitors of a given virus enzyme. In January 1998 a project proposal with the title "Protein-Protein-Docking" was submitted to the DFG Research Cluster "Informatikmethoden für die Analyse und Interpretation großer genomischer Datenmengen". After the acceptance of the proposal by the DFG the project started in October 1998. In this project, our group cooperates with the Max Planck Institute for Molecular Physiology in Dortmund and the Max Planck Institute for the "Enzymology of the Protein Folding Process" in Halle. In the proposal, the following research goals are formulated: (1) We plan to accelerate the process of structure determination for protein complexes by combining NMR and XRAY techniques with protein docking algorithms. (2) During the project the unknown 3D structures of some important protein complexes will be predicted using the docking algorithm and afterwards determined via NMR or XRAY. Within a few weeks after the start of the project we were able to implement a complex energetic evaluation function using our new software library BALL. The new energetic evaluation function improves the quality of the docking results significantly.

In Summer 1998 we started a new project called GELENA (GEne transfer systems based on LEctin-modified NAnoparticles). The goal of the project is the development of a new gene transfer method based on nanoparticles loaded with DNA vectors. The gene transfer method is intended to cure diseases, such as cystic fibrosis, caused by genetic defects. The nanoparticles will be designed and synthesized by researchers from the Institute for New Materials (INM) in Saarbrücken. Researchers of the Institute for Human Genetics of the University are working on the DNA vectors that should be transported into the corresponding cells in order to repair their genetic defects. Peptides or peptide mimetics that will identify the corresponding cells by docking reactions and that will bring the nanoparticles into contact with these cells have to be attached to the nanoparticles via spacer molecules. In close cooperation with the biotechnology company Across Barriers GmbH and the Institute for Biopharmacy and Pharmaceutical Technology of the University we are designing small peptide mimetics that have a high binding affinity to certain sugar molecules (so-called oligomers of N-acetyl-glucosamine (NAG)). We are interested in these sugar molecules because they cover the surface of epithelial cells. We are about to study and compare sugar binding sites.

Since 1995 we have been developing and implementing efficient parallel algorithms for molecular dynamics (MD) simulations of synthetic polymers, which are the base of all varieties of plastic. MD simulations are used to test hypotheses about chemical processes by simulating the motions of the atoms of a molecular system. Our MD simulation algorithms make use of the special properties and behavior of the simulated synthetic polymers. The old version of our MD simulation algorithm (that we described in the progress report from 1997) had a speedup efficiency of 77.9 % for 16 processors and a speedup efficiency of 51.4 % for 32 processors. During the last two years, we have developed new techniques like the lookahead method that enabled us to reduce the number of

communication steps and the amount of data that has to be transmitted. The new version based on these techniques has a speedup efficiency of 92.6 % for 16 processors and a speedup efficiency of 74.9 % for 32 processors. Furthermore, we gave experimental evidence that the new version has an almost optimal speedup for up to 32 processors. Since our approach is only suited for a small number of processors (< 40), we have reached the conclusion of the MD simulation project.

6.1 Computational Molecular Biology

6.1.1 Multiple Sequence Alignment

Investigators: Hans-Peter Lenhof, Kurt Mehlhorn, Petra Mutzel, and Knut Reinert

Let $S = \{S_1, S_2, \ldots, S_k\}$ be a set of k strings of lengths $n_1, \ldots, n_k$ over an alphabet Σ and let $\widehat{\Sigma} = \Sigma \cup \{-\}$, where "$-$" (dash) is a symbol to represent "gaps" in strings. An *alignment* of S is a set $\widehat{S} = \{\widehat{S}_1, \widehat{S}_2, \cdots, \widehat{S}_k\}$ of strings over the alphabet $\widehat{\Sigma}$ that satisfies the following two properties: (1) the strings in $\widehat{S}$ all have the same length, and (2) ignoring dashes, string $\widehat{S}_i$ is identical to string S_i. An alignment in which each string $\widehat{S}_i$ has length l can be interpreted as an array of k rows and l columns where row i corresponds to string $\widehat{S}_i$. Two characters of distinct strings in S are said to be *aligned* under $\widehat{S}$ if they are placed in the same column of the alignment array. A scoring function assigns to each alignment a score such that in as many situations as possible the score is in accordance with biology in the sense that alignments are assigned high scores if and only if they are *biologically meaningful*. Since sequence alignment is of central importance in the interpretation of protein and DNA sequences, a lot of research has been conducted in this area and dozens of algorithms for the different alignment problems have been published. Almost all published non-heuristic algorithms for these problems use dynamic programming. Since the time and space complexity of dynamic programming algorithms grows exponentially with the number of sequences, dynamic programming approaches can solve only rather small problem instances to optimality, even if elaborate bounding procedures are applied (see [10]).

In 1995, Kurt Mehlhorn suggested to test whether new techniques from the area of combinatorial optimization can be successfully applied to sequence alignment problems. We first studied the so-called Maximum (Weight) Trace (MT) problem, an instance of the Multiple Sequence Alignment problem introduced by John Kececioglu [5]. In the MT problem, we view the character positions of the k input strings in S as the vertex set V of a k-partite graph $G = (V, E)$ called the input *alignment graph*. The edge set E connects pairs of characters that one would like to have aligned. We call an edge in E an *alignment edge* and say that an alignment edge is *realized* by an alignment if the endpoints of the edge are placed into the same column of the alignment array. The subset of E realized by an alignment $\widehat{S}$ is called the *trace* of $\widehat{S}$. The notion of a trace of two strings is a basic concept in sequence comparison (see, for instance, [15] pp. 10–18) which Kececioglu [5] generalized to multiple sequence alignment with the notion of a trace of an alignment graph. We discovered an integer linear programming formulation of the MT problem and developed a branch-and-cut algorithm that was able to solve problem instances that cannot be handled by dynamic programming approaches [14].

In [7] we introduced the *Generalized Maximum Trace Problem* (GMT) in which we allow multiple edges between two vertices in the alignment graph G and in which we partition the edge set E into a set D of so-called *blocks*. A block is a trace in which every edge is incident to nodes in the same pair of sequences. We regard a block $d \in D$ as realized if all the edges in d are realized. Every block $d \in D$ has a weight w_d representing the benefit of realizing that block, and the weight of an alignment is the sum of the weights of the blocks it realizes. The goal is to compute an alignment

$\widehat{S}$ of maximum weight. Notice that this results in the construction of a multiple alignment out of local pairwise alignments.

Most commonly used scoring schemes are based on the similarity of single pairs of characters (see, for instance, [2] or [3]). This corresponds to a partition of the edges into singleton sets and is equivalent to the original MT formulation. It is worth noting that the singleton case includes as a special case the well studied sum-of-pairs multiple alignment problem. GMT also captures more general scoring schemes based on the similarity of pairs of whole segments of the sequence pairs (see, for instance, [1], [12], and [16]).

The graph-theoretic formulation of the GMT enabled us to give an ILP formulation for the GMT in which we associate with every block d in D a binary variable x_d that indicates whether a block is realized ($x_d = 1$) or not ($x_d = 0$). An integer solution is feasible if the alignment edges of the realized blocks form a trace. The goal is to find the feasible solution that realizes a set of blocks with maximum overall weight.

We investigated the structure of the GMT polytope $P_{\mathcal{T}}(G)$, which is defined as the convex hull of all feasible incidence vectors. This is a first essential step on the way to an efficient branch-and-cut algorithm. We were able to identify numerous classes of facet-defining inequalities and for many of these classes we could devise exact and heuristic separation algorithms that turn the theoretical knowledge about the polyhedra into practical routines for deriving upper bounds. We implemented the algorithm using the branch-and-cut framework ABACUS [4] and the Library of Efficient Datatypes and Algorithms LEDA [11]. Our implementation of the branch-and-cut algorithm for the GMT shows that the use of methods from combinatorial optimization in the field of sequence alignment leads to algorithms that are comparable or superior to existing algorithms based on dynamic programming. We can, for example, align up to 18 sequences of lengths ≈ 200, a problem size not tractable for dynamic programming based approaches.

Our original formulation was also the basis for the second alignment problem we address, the *Structural Maximum Trace Problem* (SMT). The aim is to compute an alignment that maximizes sequence and structure consensus simultaneously. To be more precise, the score that is optimized is a weighted sum of the sequence similarity and the structural similarity of the sequences under consideration. In this context, structural similarity stands for the similarity of the *secondary structures* of the sequences, which in our examples are mostly RNA sequences.

An RNA molecule is generally a single-stranded nucleic acid molecule that folds in space due to the formation of hydrogen bonds between its bases. Conventional sequence alignment algorithms can only account for the sequence and thus ignore structural aspects. In RNA molecules it is this secondary structure that carries the functionality and hence tends to be conserved through evolution. Our aim is to align the sequences using the structural information given, thereby exhibiting not only sequence similarity but also structural similarity.

In the case of the SMT problem we showed in [8, 9] that the ILP formulation for the MT problem can be extended in order to deal with structural information. This means that the input to the SMT problem can be viewed as an alignment graph, with additional edges for possible *interactions* or *base pairs* between two characters of one sequence. The list of base pairs may be produced by some secondary structure prediction program or may be a list of all possible Watson-Crick base pairs (`A-U` or `C-G`).

A structural alignment can not only realize an alignment edge, *i.e.*, the match of two characters of the sequences, but also an *interaction match*. A pair of interactions in two different sequences is said to be *aligned* or *matched* if the interacting characters in the two sequences are aligned.

We devised an ILP formulation for the SMT in which we associate with every alignment edge e in E a binary variable x_e that indicates whether the edge is realized ($x_e = 1$) or not ($x_e = 0$). For

the same purpose we assign to each interaction match m a binary variable x_m. An integer solution is feasible if the realized alignment edges form a trace and if each character is involved in at most one realized interaction match. Each variable is assigned a weight that represents the benefit of realizing the alignment edge or the interaction match. The goal is to find a feasible solution of maximal overall weight.

The investigation of the SMT polytope shows that some classes of inequalities are in essence the same as for the GMT polytope. We found three new classes of valid inequalities and showed under what conditions they are facet-defining. We implemented a branch-and-cut algorithm for structurally aligning two RNA sequences and were able to align sequences of length ≈ 1400 provably better than conventional algorithms. Indeed, to our knowledge, there is no other algorithm that is able to structurally align sequences of this length to optimality. Algorithms based on dynamic programming cannot analyze sequences longer than a few hundred nucleotides. Moreover, our algorithm can easily be extended to handle multiple sequences.

We summarized our new results in [6]. This paper is a condensed version of Knut Reinert's PhD thesis [13]. In his thesis he presents new results for both the SMT and GMT problem and derives an ILP formulation for multiple sequence alignment with arbitrary gap costs. We see the introduction of the polyhedral approach to the area of sequence alignment as a main contribution of our work. We claim that this method has plenty of room for improvement, while traditional methods based on dynamic programming are already thoroughly studied and hard to improve.

References

[1] S.F. Altschul and B.W. Erickson. Locally optimal subalignments using nonlinear similarity functions. *Bulletin Mathematical Biology*, 48:633–660, 1986.

[2] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcut. A model of evolutionary change in proteins. In M.O. Dayhoff, editor, *Atlas of Proteins Sequence and Structure*, volume 5, pages 345–352. National Biomedical Research Foundation, Washington, D.C., 1979.

[3] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Science*, 89:10915–10919, 1992.

[4] M. Jünger and S. Thienel. The design of the branch and cut system ABACUS. Technical Report 97.260, Institut für Informatik, Universität zu Köln, 1997.

[5] J. D. Kececioglu. *Exact and approximation algorithms for DNA sequence reconstruction.* PhD thesis, University of Arizona, 1991.

- [6] J. D. Kececioglu, H.-P. Lenhof, K. Mehlhorn, P. Mutzel, K. Reinert, and M. Vingron. A polyhedral approach to sequence alignment problems. *To appear in Discrete Applied Mathematics.*
- [7] H.-P. Lenhof, B. Morgenstern, and K. Reinert. An exact solution for the segment-to-segment multiple sequence alignment problem. *To appear in BIOINFORMATICS.*
- [8] H.-P. Lenhof, K. Reinert, and M. Vingron. A polyhedral approach to RNA sequence structure alignment. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB 98)*, pages 153–162, 1998.
- [9] H.-P. Lenhof, K. Reinert, and M. Vingron. A polyhedral approach to RNA sequence structure alignment. *Journal of Computational Biology*, 5(3):517–530, 1998.
- [10] M. Lermen and K. Reinert. The practical use of the $\mathcal{A}^*$ algorithm for exact multiple sequence alignment. Research Report MPI-I-97-1-028, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, December 1997.

[11] K. Mehlhorn and S. Näher. LEDA, a platform for combinatorial and geometric computing. *Communications of the ACM*, 38(1):96–102, 1995.

[12] B. Morgenstern, W.R. Atchley, K. Hahn, and A. Dress. Segment-based scores for pairwise and multiple sequence alignments. In *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB-98), (in press)*, 1998.

- [13] K. Reinert. *A polyhedral approach to sequence alignment problems.* PhD thesis, Universität des Saarlandes, 1999.

[14] K. Reinert, H.-P. Lenhof, P. Mutzel, K. Mehlhorn, and J. Kececioglu. A branch-and-cut algorithm for multiple sequence alignment. In *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB 97)*, pages 241–249, 1997.

[15] D. Sankoff and J.B. Kruskal. *Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison.* Addison Wesley, 1983.

[16] W.J. Wilbur and D.J. Lipman. The context dependent comparison of biological sequences. *SIAM Journal Applied Mathematics*, 44(3):557–567, 1984.

6.1.2 QUASAR: Q-gram based database searching using a Suffix ARray

Investigators: Stefan Burkhardt, Andreas Crauser, Paolo Ferragina, and Hans-Peter Lenhof

Large databases holding DNA and protein sequences are nowadays a cornerstone of molecular biology. Whenever a new gene is sequenced, searching in the appropriate databases for similar sequences is the first step to acquire new knowledge about the function and the relationship of the gene. Thus, efficient algorithms for approximate string matching play an important role in the area of computational molecular biology. These algorithms can also be applied to cluster similar sequences into sequence families [7]. The sequence assembly problem is another important application of sequence database search: A large number of short subsequences of a new DNA molecule are sequenced by robots. The subsequences are compared in order to find overlaps. Since the sequencing process produces errors in the data, algorithms for exact string matching are not able to detect all overlaps. Therefore the overlap detection has to be carried out with algorithms for similarity search. Using the information about overlaps between subsquences, the sequence of the entire DNA molecule can be computed. Another application stems from the design of expression arrays. Expression arrays are diagnostic tools for determining which genes are active (expressed) in certain cells. For example, expression arrays of human genes can be used to analyze the metabolic differences of cancer and normal cells. The selection of representative clones for an expression array is based on the clustering of Expressed Sequence Tags (EST). An EST is a DNA fragment of length around 500 that stems from an expressed gene. The clustering problem can be defined as follows: Given a database of ESTs, determine for each EST all other ESTs in the database that stem from the same gene, *i.e.*, determine the cluster for each EST. The clusters can be computed in the following way: Determine all pairs of overlapping ESTs in the database and store this overlap information in a graph in which the nodes represent the ESTs and the edges the overlaps. By analyzing the connected components of this overlap graph, the clusters and the representative clones can be computed. Because of errors in the EST data, again algorithms for similarity search have to be used.

There are well known programs for similarity search in sequence databases. The most commonly used programs are BLAST [1] and FASTA [11]. Especially BLAST is impressively fast. For a given query string, BLAST performs a linear scan of the whole collection of sequences in the database and detects all local similarities. Like BLAST, most search engines scan the whole database linearly. Since the databases are growing exponentially, more sophisticated searching tools have to be developed to handle the computational challenges arising in new applications. For example, in the

sequence assembly problem and in the EST clustering problem all-against-all comparisons of the database sequences have to be carried out.

In the field of exact string matching, indexing data structures and efficient algorithms for building and preprocessing these data structures and for searching in them have been developed. When searching for a query pattern in such a data structure, only small parts of the text (database) actually need to be explicitly accessed. Only a few attempts have been made to adapt these indexing techniques to the similarity searches needed for the presented biological problems. Martinez [9] gave the first application of a position tree in molecular biology. This data structure requires about 16 times the space needed to store the original data. An index structure of similar size was published by Heumann [5]. The size of these data structures may create serious problems when applied to large data collections. Myers [10] suggested a sub-linear search algorithm that is centered around an index built on small substrings of the database sequences. The IBM product FLASH [4] takes advantage of a large "probabilistic" index over randomly chosen substrings. They report an 18 GB index for a 100 million residue database which makes such an approach impractical for large databases.

In cooperation with the German Cancer Research Center in Heidelberg we created QUASAR, an algorithm designed for multiple high similarity searches in DNA databases. This algorithm is based on a filter technique introduced by Jokinen und Ukkonen [6]. It requires locating exact matching substrings in the database. We achieve this by using a suffix array [8] of the database which allows us to do exact matching without scanning the whole database. The combination of these two ideas results in a very efficient filter algorithm that returns hot-spots, *i.e.*, for a given query it returns possible locations of approximate matches in the database. These hot-spots are then examined more closely using an alignment algorithm.

The algorithm together with the results of some "real world" experiments have been published in [2, 3]. QUASAR achieves a significant speedup over currently used search algorithms like BLAST and FASTA. With appropriate values for the algorithmic parameters, speed increases of two orders of magnitude are possible. We also implemented a secondary memory version of our algorithm that achieves almost the same speedup factor. The algorithm enables researchers to cluster much larger EST databases. In a first test, the Mouse EST database containing roughly 200,000 ESTs was clustered in less than 10 hours on a Sun UltraSparc 2 with 1 GB of main memory. Although designed for high similarity search, our algorithm showed the same sensitivity as BLAST for more than 94 % of 2000 test queries.

References

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

- [2] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array (QUASAR). Technical Report MPI-I-98-024, Max-Planck-Institut für Informatik, 1998.
- [3] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array (QUASAR). In *Proc. of the Third Annual International Conference on Computational Molecular Biology (RECOMB 99)*, pages 77–83, 1999.

[4] A. Califano and I. Rigoutsos. FLASH: A fast look-up algorithm for string homology. In L. Hunter, D. Searls, and J. Shavlik, editors, *Proc. of the 1st International Conf. on Intelligent Systems for Molecular Biology*, pages 56–64, 1993.

[5] K. Heumann. *Biologische Sequenzdatenanalyse großer Datensätze basierend auf Positionsbaumvarianten.* PhD thesis, Technische Universität München, Institut für Informatik, 1997.

[6] P. Jokinen and E. Ukkonen. Two algorithms for approximate string matching in static texts. In *Proc. of the 16th Symposium on Mathematical Foundations of Computer Science*, volume 520 of *Lecture Notes in Computer Science*, pages 240–248, 1991.

[7] A. Krause and M. Vingron. A set-theoretic approach to database searching and clustering. *Bioinformatics*, 14:430–438, 1998.

[8] U. Manber and G.W. Myers. Suffix arrays: A new method for on-line string searches. In *Proc. of the first annual ACM-SIAM Symposium on Discrete Algorithms*, pages 319–327, 1990.

[9] H.M. Martinez. An efficient method for finding repeats in molecular sequences. *Nucleic Acids Research*, 11(13):4629–4634, 1983.

[10] E.W. Myers. A sublinear algorithm for approximate keyword searching. *Algorithmica*, 12(4/5):345–374, 1994.

[11] W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *PNAS*, 85:2444–2448, 1988.

6.1.3 BALL: Biochemical ALgorithms Library

See Section 8.3 on page 138.

6.1.4 Protein Docking

Investigators: Oliver Kohlbacher and Hans-Peter Lenhof

In the next century, virtual labs will play a key role in the life sciences. Modules for simulating chemical reactions will be an important component of these virtual labs. We are developing and implementing algorithms to predict reactions between proteins, *i.e.*, the algorithms predict the 3D structures of the protein complexes and compute the stability of these complexes. This problem is called the Protein Docking (PD) problem. An abstract version of the PD problem can be formulated as follows: Given two proteins A and B and their 3D structures, compute the 3D structures of possible protein complexes AB. A protein docking algorithm outputs a list of possible docking complexes AB sorted with respect to a certain fitness or energy function that measures the stability of the complexes.

Most protein docking algorithms use the following approach (see, for example, [10, 2, 8, 9, 1]): First, a set of possible docking conformations is computed that hopefully contains an approximation of the conformation found in nature. Second, the geometric fitness values of the conformations are calculated. The geometric fitness function measures the geometrical complementarity of the candidates, *i.e.*, it measures whether there is a good fit between the surfaces of the docking partners. Only the candidates with high geometric fitness values are evaluated with respect to their chemical complementarity. Thus, the geometric fitness function serves as a first filter to reduce the number of candidates. The remaining conformations are tested with different chemical filters. At the end, the algorithm outputs a short list of possible conformations that are sorted with respect to a special fitness function or with respect to energy values (see, for example, [1]). The first list element stores the conformation that has the best fitness or energy value.

The most important criterion for judging protein docking algorithms is the ranking criterion. A protein docking algorithm successfully predicts the 3D structure of a protein complex if the first element (or at least one of the first elements) of the result list is a good approximation of the natural complex structure. The quality of the approximation can be measured by comparing all atom coordinates of the natural complex and the predicted complex and by calculating the root

mean square deviation of the atom coordinates. Of course, time and space complexity are also important criteria.

In 1995, we presented a parallel docking algorithm [5] that uses a new scheme to generate the candidate conformations. The second new ingredient of this algorithm was a geometric scoring function that counts the number of van der Waals contacts between atoms of A and atoms of B. Tests with "real world" docking examples showed that the algorithm delivers high quality rankings, it is fast, and the parallel version has a good speedup. In [6, 7] we presented a new chemical scoring function for judging the chemical fitness of conformations. The new fitness measure is also based on the van der Waals contact principle: We consider all atom pairs that have a van der Waals contact, but instead of adding a constant for each atom pair (a, b), we add a chemical weight that depends on the atom pair. The weights are calculated by making a statistical analysis of van der Waals contacts in known protein complexes. Our current docking algorithm uses this chemical scoring function as a second filter after the geometric fitness function. The conformations in the result list of the algorithm are sorted with respect to the sum of two fitness functions. Tests with the combined fitness function showed that the rankings were significantly better than the rankings with the pure geometric fitness function, but the running times increased significantly. We also compared our results with the results of well known docking algorithms. The rankings of our results were better than the rankings of these algorithms.

In January 1998 we submitted a project proposal with the title "Protein-Protein-Docking" to the DFG Research Cluster "Informatikmethoden für die Analyse und Interpretation großer genomischer Datenmengen". After the approval by the DFG, the project started in October 1998. In this project our group cooperates with two other groups of researchers: a group from the Max Planck Institute for Molecular Physiology in Dortmund that works in the area of XRAY diffraction and a group from the Max Planck Institute for the Enzymology of the Protein Folding Process in Halle that specializes in structure determination with NMR techniques. The main goal of the project is the continued development and expansion of the protein docking software. New complex energetic evaluations will be added. Another new component will be implemented that calculates certain NMR spectra of the predicted potential complexes and compares the calculated and measured spectra. In the course of the project, the unknown 3D structures of some important complexes will be predicted with the docking algorithm and their structure will be determined by XRAY or NMR techniques. The predicted structures will be used to accelerate the process of structure determination by speeding up the solution of the phase problem (in the Fourier synthesis of the XRAY diffraction method) and the assignment of NMR shifts to atoms. The comparison of predicted and measured structures will exhibit the strong and weak points of our docking approach and will help to improve the docking algorithm, especially the energetic evaluations that are used to rank the last remaining candidates. We will also work on the design of cyclic peptide mimetics that inhibit the docking reaction of the two monomers that build the reverse transcriptase of the HIV virus. The research group in Dortmund will design a few hundred cyclic peptides whose docking energies will be predicted with our docking algorithm. Using a biochemical test, the best peptides will then be identified in Dortmund. The summary about protein docking algorithms and energetic evaluations in our project proposal will be published as an overview article [4]. The number of publications therein shows the rapid development in this field.

Thanks to our software library BALL, we were able to implement two complex energetic evaluation functions within a few weeks after the start of the project. The first function is a combination of an energetic contact measure that has been developed by Zhang, *et al.* [11] and the electrostatic interaction energy of the two proteins calculated with the Poisson-Boltzmann approach. The second energetic evaluation function is due to Jackson and Sternberg [3]. It splits the total binding free

energy ΔG_{total} into three components: ΔG_{solv}, the change in solvation free energy of both partners during the association, ΔG_{int}, the electrostatic interaction energy between A and B, and ΔG_{cav}, the cavitation free energy caused by the change in the molecular surface during the association (used to describe the hydrophobic interaction quantitatively).

We carried out a certain number of docking experiments where we used these energetic evaluation functions to rank the complex conformations that passed our simple geometric and chemical filter. Both scoring functions improved the rankings significantly. The energetic function of Jackson and Sternberg yields especially good rankings. In [6] we presented docking results for experiments with unbound conformations of A and B. Among the considered docking complexes were three trypsin complexes whose complex conformations had not been (optimally) predicted by our old algorithm. By using the energy function of Jackson and Sternberg for the ranking of the remaining conformation, the docking algorithm was able to predict the complex conformation in all three examples, *i.e.*, the first elements of the result lists were good approximations of the natural complex conformations.

The development of methods and software for the simulation of nuclear magnetic resonance spectra is another important part of the project. The simulation of these spectra is important because it promises to speed up the laborious shift assignment during the structure determination by using nuclear magnetic resonance methods. A very specialized type of spectrum is the heteronuclear (^{1}H-^{15}N) HSQC spectrum that results from the N-H bonds in the peptide bond. This type of spectrum is relatively easy to obtain and contains information on the backbone torsional angles of the protein. A high-quality prediction of these spectra permits the validation or falsification of structural models obtained from docking experiments. We are currently developing methods to predict these spectra. Our approach separates the total (secondary) shift of each nucleus into different contributions caused by ring currents (due to aromatic rings in the neighborhood of the nuclei), the electrostatic field, and the magnetic anisotropy of neighboring bonds. We are currently calculating each of these contributions for small model systems using quantum chemical approaches (density functional theory and post-Hartree-Fock ab-initio methods). Experimental verification of these theoretical methods is crucial, thus we closely collaborate with Peter Bayer in Halle, who provides the measured ^{15}N shifts for small systems. We identified the MP2/6-31+G* method as a reliable, though computationally very expensive, method to predict the ^{15}N shifts of small compounds. Using these data, we will now develop empirical expressions that allow a fast approximate prediction of these shift contributions. These methods will then become a further component of BALL.

References

[1] D. Fischer, S. L. Lin, H. J. Wolfson, and R. Nussinov. A geometry-based suite of molecular docking processes. *J. Mol. Biol.*, 248:459–477, 1995.

[2] D. Fischer, R. Norel, R. Nussinov, and H. J. Wolfson. 3-D docking of protein molecules. In *Lecture Notes in Computer Science 684*, pages 20–34. Springer Verlag, New York, 1993.

[3] R.M. Jackson and M.J.E. Sternberg. A continuum model for protein-protein interactions: Application to the docking problem. *J. Mol. Biol.*, 250:258–275, 1995.

- [4] O. Kohlbacher and H.-P. Lenhof. New developments in protein-protein docking. To appear in Macromol. Theory and Simulation, 1999.

[5] H.-P. Lenhof. An algorithm for the protein docking problem. In Dietmar Schomburg and Uta Lessel, editors, *Bioinformatics: From nucleic acids and proteins to cell metabolism. GBF Monographs Volume 18*, pages 125–139, 1995.

[6] H.-P. Lenhof. New contact measures for the protein docking problem. In *Proc. of the First Annual International Conference on Computational Molecular Biology RECOMB 97*, pages 182–191, 1997.

- [7] H.-P. Lenhof. Parallel Protein Puzzle: A new suite of protein docking tools. Technical Report 44, GWDG, 1997.

[8] S. L. Lin, R. Nussinov, D. Fischer, and H. J. Wolfson. Molecular surface representations by sparse critical points. *Proteins*, 18:94–101, 1994.

[9] R. Norel, S. L. Lin, H. J. Wolfson, and R. Nussinov. Shape complementarity at protein-protein interfaces. *Biopolymers*, 34:933–940, 1994.

[10] H. Wang. Grid-search molecular accessible surface algorithm for solving the docking problem. *J. Comput. Chem.*, 12:746–750, 1991.

[11] C. Zhang, G. Vasmatzis, J. L. Cornette, and C. DeLisi. Determination of atomic desolvation energies from the structures of crystallized proteins. *J. Mol. Biol.*, 267:707–726, 1997.

6.2 GELENA: Non-viral GEntransfer systems based on LEctin modified NAnoparticles

Investigators: Oliver Kohlbacher and Hans-Peter Lenhof

GELENA is an interdisciplinary project of the Institute for Biopharmacy and Pharmaceutical Technology of the University of Saarbrücken, the Institute for New Materials (INM) in Saarbrücken, the Institute for Human Genetics of the University of Saarbrücken, the biotechnology company Across Barriers, and our group. The project goal is the development of a new non-viral gene transfer system. Non-viral gene transfer systems are of high interest, as viral transfer systems have shown some serious disadvantages in first experiments.

Lectins are sugar-binding proteins that occur in plants like tomatoes, wheat, peas, and many more. They bind sugars that also occur on the outer surface of epithelial cells, which are the favorite target for a gene therapy of different diseases (*e.g.*, cystic fibrosis). We intend to design a transfer system that consists of inorganic nanoparticles (sub-micrometer particles of inorganic oxides). These nanoparticles are then modified with lectins on their surface. This modification should cause the particles to bind to the cell surface and to enter the cell, because the lectins bind to the sugars on the cell's surface. For a gene therapy, these particles are loaded with the DNA needed to "repair" the cell.

Our first objective in this project is the analysis of the binding mode of lectins to sugars. Using this knowledge, we will then design modified lectins. These modified lectins should be smaller than the existing lectins in order to reduce the immune response. Using our tools developed for protein docking, we will predict the binding constants of modified lectins, thus selecting the most promising candidates for a synthesis.

First results on the sugar binding mode of lectins have been presented at the Annual Meeting of the American Association of Pharmaceutical Scientists 1998 [1].

References

- [1] D. Neumann, E. Haltner, C.-M. Lehr, O. Kohlbacher, and H.-P. Lenhof. Investigating the sugar-lectin interaction by computational chemistry: tunneling the epithelial barrier. In *Abstracts of the AAPS Annual Meeting, San Francisco, USA*, page 549. American Association of Pharmazeutical Scientists, November 1998.

6.3 Computational Chemistry

6.3.1 Molecular Dynamics Simulation for Synthetic Polymers

Investigators: Hans-Peter Lenhof, Peter Müller and Christine Rüb

Molecular dynamics (MD) simulations have become an important tool for testing hypotheses about chemical and physical processes. In an MD simulation, the motions of the atoms of a molecular system are simulated using classical mechanics. Given the atomic positions and velocities at time t, interaction forces are calculated according to a physical model (the force field). Then, by numerical integration of Newton's equations of motion, the positions and velocities of the atoms at time $t + \Delta$ can be computed. The time step Δ must be chosen carefully and is typically on the order of 1 fs ($= 10^{-15}$s). Unfortunately, simulation periods of up to milliseconds or even seconds are desirable. Such tiny time steps mean that the above mentioned force field evaluation and integration — which constitute one iteration in an MD simulation — must be executed very often, imposing a heavy drain on computing resources.

One promising way to accelerate MD simulations is using parallel computers. Over the last 10 to 15 years, many parallel algorithms have been developed and implemented (*e.g.*, [2, 7, 8, 6]; a comprehensive overview can be found in [1]). Most of these algorithms have been designed for the simulation of proteins. Proteins fold up to compact structures with little overall dynamics whereas the objects of our simulations — synthetic polymers — show a completely different behavior. These macromolecules form long, loose coils and their typical trajectories are three–dimensional random walks with lots of movement. Furthermore, the dynamics of such a polymer can be studied by simulating only one single polymer chain; surrounding atoms of a solvent or other polymer molecules need not be included explicitly as their influence can be modeled by stochastic forces (for more details, see [4]). Hence, parallel algorithms that were developed for proteins are not efficient for synthetic polymers.

We have therefore developed and implemented parallel algorithms that take the special properties and behavior of synthetic polymers into consideration. These algorithms are platform independent and run on any parallel architecture with a moderate number of processors, distributed memory, and message passing as the means of communication.

Our approach uses a straightforward molecule decomposition, *i.e.*, the polymer chain is divided into segments of roughly the same size, which are allocated to the processors. Most of the necessary data exchanges then occur between neighboring processors ("neighboring" with respect to the subdivision of the chain) and can be largely hidden by non–blocking send/receive operations, *i.e.*, communication is going on while the processor itself can do useful local work. This communication is regular and concerns always the same atoms. In contrast to this, quickly changing and highly irregular communication patterns arise between some processors due to bends and loops of the chain. In principle, each processor must know all other atoms of the entire chain, requiring expensive global communication. In order to deal with this, we use a coarse–grained method. Each processor puts its segment of the chain into several bounding boxes. These boxes are exchanged and checked for intersections. Only atoms within these intersection areas must actually be sent. We investigated two variants. In method A, all boxes are sent to a master processor, which does the overlap tests and sends the results back. In method B, each processor broadcasts its boxes to all other processors and does the overlap testing itself. The expensive broadcast operation has to be done only at the beginning of the simulation. Most pairs of processors will find out that their bounding boxes do not overlap. In that case, a separating plane is calculated and used for testing. As long as the boxes of a processor do not intersect this plane, the box exchange with the corresponding partner can be suspended. For moderate numbers of processors (≤ 32), both variants are almost

equivalent and show good parallel efficiency. For further improvement, we applied two dynamic load balancing techniques. The first one is based on a diffusive method where the responsibility for little pieces of the chain is shifted between neighboring processors. The other one tries to balance each processor's load over time by linking the amount of work due to random number generation to the amount of waiting times. Both methods turned out to be roughly equivalent, leading to only slight improvements of running times (about 3 % gain in parallel efficiency for 32 processors, see [5, 9]).

In order to find out how much speedup of our parallelization approach can be expected at best, a simulation of the actual MD simulation was created. In this simulation, all work is perfectly balanced among the processors. Furthermore, each processor is given the same average amount of communication (number of messages and lengths). These averages were obtained from measurements on real MD simulation runs. The results in Table 6.1 show that our algorithms are very close to these upper bounds for up to 16 processors and quite close for a higher number of processors.

p	max. efficiency	achieved
2	98.7 %	98.2 %
4	97.9 %	97.3 %
8	95.8 %	95.3 %
16	92.9 %	92.6 %
24	86.2 %	80.0 %
32	78.3 %	74.9 %

Table 6.1: Speedup bounds for the simulation of a polyethylene chain (3002 atoms) on Cray–T3E.

An often neglected aspect of developing MD simulation algorithms is the verification of simulation results. We attached great importance to this in order to see whether the computed output matches known theory, thus proving the correctness and usefulness of the implementation [3]. These efforts also gave rise to extensive research on random number generators which are a key component in our algorithms (see the following section).

From January 1995 to December 1997, this project was supported by the DFG Research Cluster "Efficient Algorithms For Discrete Problems and Their Applications", grants YE 952/1–1,1–2.

References

[1] D. Beazley, P. Lomdahl, N. Grønbech-Jensen, R. Giles, and P. Tamayo. Parallel algorithms for short–range molecular dynamics. *World Scientific's Annual Reviews in Computational Physics*, 3, 1995.

[2] H. Heller, H. Grubmüller, and K. Schulten. Molecular dynamics simulation on a parallel computer. *Molecular Simulation*, 5:133–165, 1990.

[3] B. Jung, H.-P. Lenhof, P. Müller, and C. Rüb. Langevin dynamics simulations of macromolecules on parallel computers. *Macromolecular Theory and Simulation*, 6:507–521, 1997.

[4] B. Jung, H.-P. Lenhof, P. Müller, and C. Rüb. Parallel MD-simulations of synthetic polymers. In *Proc. of the 8th SIAM Conference on Parallel Processing for Scientific Computing*, 1997.

- [5] B. Jung, H.-P. Lenhof, P. Müller, and C. Rüb. Simulating synthetic polymer chains in parallel. In *Proc. of the 7th International Conference on High Performance Computing (HPCN), Amsterdam, LNCS 1593*, pages 13–22. Springer, April 1999.

[6] M. Nelson, W. Humprey, A. Gursoy, A. Dalke, L. Kale, R. Skeel, and K. Schulten. NAMD - A parallel object–oriented molecular dynamics program. *Journal of Supercomputing Applications and High Performance computing*, 10:251–268, 1996.

[7] W. Smith and T.R. Forester. Parallel macromolecular simulations and the replicated data strategy. *Computer Physics Communications*, 79:63–77, 1994.

[8] A. Windemuth. Advanced algorithms for molecular dynamics simulation: The program PMD. In T. G. Mattson, editor, *Parallel Computing in Computational Chemistry*, pages 151–168. ACS Books, 1995. http://tincan.bioc.columbia.edu/Lab/pmd/.

- [9] P. Müller. *Parallel Molecular Dynamics Simulations for Synthetic Polymers.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.

6.3.2 On the Quality of Random Number Generators for the Normal Distribution

Investigator: Christine Rüb

While trying to verify our simulation program for synthetic polymers (see above), it became apparent that the uniform number generator for the normal distribution we were using did not deliver the desired output. This has motivated us to investigate the quality of such random number generators. Unlike the case of uniform random number generators, there exist only a few papers on the quality of random number generators for other distributions.

Most random number generators for the normal distribution work by transforming one or more variates from the uniform distribution to one or more variates from the normal distribution. That is, one is given a transformation algorithm that is used together with a uniform random number generator. The transformation algorithms are exact, which means that the produced output will be normally distributed and independent provided that the input values are uniformly distributed and independent and that all arithmetic is carried out with infinite precision. However, none of these assumptions is fulfilled in practice and sometimes this will be reflected by the output of a transformation algorithm. In fact, the problems we encountered in our polymer simulation program could be traced back to long–range correlations of the uniform random number generator used.

In [8], we present results of a study we performed on how strongly properties (like defects or a limited number of bits used for the output) of the uniform number generator affect the output of the transformation algorithm. Our test suite contained 13 transformation algorithms, ranging from the long-known Box-Muller algorithm [1] to newer algorithms like the Transformed Rejection algorithm [4]. We used 17 uniform random number generators ranging from generators with known defects like r250 [5] to modern, high quality generators like the Mersenne Twister [6]. The test procedures used were mainly standard statistical tests like tests for the first four moments of the normal distribution and χ^2 tests. Additionally, we tested the sensitivity of the transformation algorithms to long–range correlations of the uniform random number generator. We also studied the effects that a small number of bits used for the uniform variates has on the output of the transformation algorithms.

The first result of this study came as a surprise: for seven of the 13 transformation algorithms tested, either the algorithm or the published program listing contains some errors without this being mentioned in the literature (for one of the algorithms, this has changed in the mean time). Some of these errors are easily detected, for example, one of the programs could not be compiled. Other errors are very subtle like the one in the algorithm Grand [2] that only becomes apparent when the uniform random number generators rounds its output values in a certain way.

The results of the standard tests showed a large variety in the sensitivity of the transformation algorithms to properties of the uniform random number generators: some transformation algorithms

seem to reflect any problem of the uniform variates while for others only very few combinations of test, (a large) repetition time, and uniform random number generator leads to a failure. It turned out that, perhaps not surprisingly, the way the uniform variates are used in the transformation algorithm affects this sensitivity greatly.

Most uniform random number generators work internally with integers. This means that their output will lie on a grid with, in most cases, a distance of at least 2^{-32} between adjacent numbers. Some applications will be sensitive to this property and there are tests like the Spacings test that are able to detect this. Transformation algorithms, on the other hand, work in general with floating point numbers, which allow for a finer distribution of the output. In fact, some of the transformation algorithms tested produce a much finer distributed output than the underlying uniform random number generator. This depends again on the way the uniform random number generators are used.

The third point addressed was the sensitivity of the transformation algorithm to higher dimensional long-range correlations of the uniform random number generator. This is mainly a problem of Linear Congruential Generators (LCGs). However, it has sometimes turned out that a differently presented uniform random number generator is, in fact, an LCG in disguise. All transformation algorithms tested show a sensitivity to two-dimensional, but also to four- or higher dimensional correlations of LCGs, which might be a problem for certain applications. Two-dimensional long-range correlations of LCGs have been considered before [3].

The paper [8] can also serve as an introduction to transformation algorithms for the normal distribution. It explains most of the known transformation techniques and the transformation algorithms used here (most books on this topic contain only very few algorithms and concentrate on the techniques). This paper also lists the known effects some properties of uniform number generators have on the output of the transformation algorithms and explains some previously unknown such effects.

Unlike most known algorithms, the algorithm Fastnorm by Wallace [10] does not transform uniform variates but works directly on normal variates. This means that at the beginning, a pool of normal variates is generated by a traditional algorithm. In every pass, this pool of numbers is then transformed into a pool of new variates, which leads to a very fast algorithm. Unfortunately, there are some problems with this method. In [7] it was shown that the algorithm originally proposed by Wallace (see [9]) will lead to defective output if consecutive numbers produced by the algorithm are added (in other words, the output variates are not independent). This defect has been traced back to the simple way the pool of old variates is scanned in a pass. In the mean time, Wallace has developed an improved version [9] that does not show the strong deviations from the expected behavior as the old version. However, there are still some smaller deviations (of the size of $1/S$ % where S is the size of the pool) that seem to be difficult to get rid of. This means that Fastnorm has more of the nature of an approximation algorithm.

References

[1] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29:610–611, 1958.

[2] R. P. Brent. A Gaussian pseudo-random number generator. *Communications of the ACM*, 17(12):704–706, December 1974.

[3] A. De Matteis and S. Pagnutti. Parallelization of random number generators and long-range correlations. *Numerische Mathematik*, 53(5):595–608, 1988.

[4] W. Hörmann and G. Derflinger. The transformed rejection method for generating random variables, an alternative to the ratio of uniforms method. *Communications in Statistics: Simulation and Computation*, 23(3):847–860, 1994.

[5] S. Kirkpatrick and E. P. Stoll. A very fast shift-register sequence random number generator. *Journal of Computational Physics*, 40:517–526, 1981.

[6] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, 1998.

- [7] C. Rüb. On Wallace's method for the generation of normal variates. Technical Report MPI–I–98–1–020, Max–Planck–Institut für Informatik, Saarbrücken, Germany, 1998. Submitted for publication in the ACM Transactions on Mathematical Software.
- [8] C. Rüb. On the quality of some random number generators for the normal distribution. Technical report, Max–Planck–Institut für Informatik, Saarbrücken, Germany, 1999. In preparation.

[9] C. S. Wallace. Source files are available by anonymous ftp at ftp.cs.monash.edu.au in directory pub/csw.

[10] C. S. Wallace. Fast pseudorandom generators for normal and exponential variates. *ACM Transactions on Mathematical Software*, 22(1):119–127, March 1996.

7 Graph Drawing

About $4\frac{1}{2}$ years ago, our group started to build a graph drawing subgroup. The new group has grown quickly by getting public and industrial funds for Ph.D. students and by attracting masters students. The success can be measured by publications in conferences and journals as well as active cooperation with industrial users. Members of our group have been invited to write survey articles [3, 6, 22], to join the program committee of the international graph drawing conferences GD '97 and GD '99, and one has been invited to be a guest editor for the special issue of the Journal of Graph Algorithms and Applications (JGAA) on the Graph Drawing Symposium 1997. Moreover, we have been involved in the organization of the graph drawing contests 1998 and 1999, which are held during the annual graph drawing symposia [5].

Some of the work described in the last report has been accepted for publication in the meantime [26, 21]; other work that was previously designated as "to appear" has been published [17, 25, 16]. However, here we will report on only our new research done during the last two years.

The methods we have used are mainly integer programming techniques [12, 21, 28, 27, 19] for NP-hard combinatorial optimization problems and pure combinatorial graph algorithms [15, 13, 14, 2, 10]. Often, planarity questions play an important role [2, 27, 8, 10, 15, 13, 14]. Moreover, we have developed a polynomial time approximation algorithm for the planar augmentation problem [8]. Our theoretical work is often transferred into software; hence, we have also written some papers on software [11, 1, 24].

We had an interesting new experience with the production of our first video on graph drawing [23], in which we try to explain the current and past research on planar straightline graph drawings. Currently, Springer-Verlag is interested in publishing the video. Recently, four members of the MPI attended the GI-Forschungsseminar on Graph Drawing for which they have written survey articles on certain topics [7, 29, 4, 9].

All our algorithms are implemented using LEDA [20] and some using ABACUS [18]. We are distributing our implementations of data structures, tools and algorithms in form of AGD, our library of **A**lgorithms for **G**raph **D**rawing, via the Internet (`http://www.mpi-sb.mpg.de/AGD/`) for non-commercial use. The company Algorithmic Solutions GmbH is distributing AGD for commercial use. Besides this, we cooperate with outside researchers, partly in industry, on various practical projects.

Graph drawing methods can roughly be classified into methods using planarization, hierarchical methods, force-directed methods, orthogonal methods, dynamical methods, clustered methods, and three-dimensional methods. So far, we have not worked on dynamical methods, force-directed methods or three-dimensional methods. We have started to do some research on clustered methods, but so far we have not published anything there. Hence, this report will concentrate on our new work on planarization methods including orthogonal methods and hierarchical methods.

References

- [1] D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD-Library: A library of algorithms for graph drawing. In G. F. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE '97)*, 1997. Venice, Italy, Sept. 11-13, http://www.dsi.unive.it/~wae97/.

- [2] T. Biedl, M. Kaufmann, and P. Mutzel. Drawing planar partitions II: HH-drawings. In J. Hromkovic and O. Sykora, editors, *Proceedings of the 24-th Workshop on Graph-Theoretic Concepts in Computer Science (WG '98)*, volume 1517 of *Lecture Notes in Computer Science*, pages 124–136. Springer-Verlag, 1998.

- [3] F. J. Brandenburg, M. Jünger, and P. Mutzel. Algorithmen zum automatischen Zeichnen von Graphen. *Informatik Spektrum*, 20(4):199–207, 1997.
- [4] R. Brockenauer and S. Cornelsen. Clusters and hierarchies. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation. Presented at the GI-Seminar *Zeichnen von Graphen*, April 6-9, Schloß Dagstuhl, Germany.
- [5] P. Eades, J. Marks, P. Mutzel, and S. North. Graph drawing contest report. In S. H. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 423–435. Springer-Verlag, 1998.
- [6] P. Eades and P. Mutzel. Graph drawing algorithms. In M. Atallah, editor, *CRC Handbook of Algorithms and Theory of Computation*, chapter 9, pages 9–1–9–26. CRC Press, 1999.
- [7] M. Eiglsperger, S. Fekete, and G. W. Klau. Orthogonal graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation. Presented at the GI-Seminar *Zeichnen von Graphen*, April 6-9, Schloß Dagstuhl, Germany.
- [8] S. Fialko and P. Mutzel. A new approximation algorithm for the planar augmentation problem. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pages 260–269, San Francisco, California, 1998. ACM Press.
- [9] R. Fleischer and C. Hirsch. Applications of graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation. Presented at the GI-Seminar *Zeichnen von Graphen*, April 6-9, Schloß Dagstuhl, Germany.
- [10] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In S. H. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 167–182. Springer-Verlag, 1998.
- [11] C. Hundack, P. Mutzel, I. Pouchkarev, and S. Thome. *ArchE*: A graph drawing system for archaeology. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 297–302. Springer-Verlag, 1997.
- [12] M. Jünger, E. Lee, P. Mutzel, and T. Odenthal. A polyhedral approach to the multi-layer crossing number problem. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 13–24. Springer-Verlag, 1997.
- [13] M. Jünger, S. Leipert, and P. Mutzel. Pitfalls of using PQ-trees in automatic graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204. Springer-Verlag, 1997.
- [14] M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In S. H. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer-Verlag, 1998.
- [15] M. Jünger, S. Leipert, and P. Mutzel. A note on computing a maximal planar subgraph using PQ-trees. *IEEE Transactions on Computer-Aided Design*, 17(7), 1998.
- [16] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications (JGAA) (`http://www.cs.brown.edu/publications/jgaa/`)*, 1(1):1–25, 1997.
- [17] M. Jünger, P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of a minor-excluded class of graphs. *Discrete Mathematics*, 182:169–176, 1998.
- [18] M. Jünger and S. Thienel. Introduction to ABACUS - A Branch-And-CUt System. *Technical Report No. 97.263, Institut für Informatik, Universität zu Köln*, 1997. To appear in Operations Research Letters, 1999.

- [19] G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

  [20] K. Mehlhorn and S. Näher. LEDA: A platform for combinatorial and geometric computing. *Comm. Assoc. Comput. Mach.*, 38:96–102, 1995.

- [21] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. *SIAM Journal on Optimization*, 1999. To appear.

- [22] P. Mutzel. Optimization in leveled graphs. In P. M. Pardalos and C. A. Floudas, editors, *Encyclopedia of Optimization*. Kluwer Academic Publishers, 1999. To appear.

- [23] P. Mutzel and R. Brockenauer. Straightline planar graph drawing. Video, 1998.

- [24] P. Mutzel, C. Gutwenger, R. Brockenauer, S. Fialko, G. Klau, M. Krüger, T. Ziegler, S. Näher, D. Alberts, D. Ambras, G. Koch, M. Jünger, C. Buchheim, and S. Leipert. AGD: A library of algorithms for graph drawing (poster-abstract). In S. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 456–457. Springer-Verlag, 1998.

- [25] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs and Combinatorics*, 14:59–73, 1998.

- [26] P. Mutzel and R. Weiskircher. Two-layer planarization in graph drawing. In K.-Y. Chwa and O. H. Ibarra, editors, *Algorithms and Computation, 9th International Symposium, ISAAC '98*, volume 1533 of *Lecture Notes in Computer Science*, pages 69–78, Taejon, Korea, 1998. Springer-Verlag.

- [27] P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

- [28] P. Mutzel and T. Ziegler. The constrained crossing minimization problem – a first approach. In P. Kall and H.-J. Lüthi, editors, *Operations Research Proceedings 1998*, pages 125–134. Springer-Verlag, 1999.

- [29] R. Weiskircher. Planar graphs. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation. Presented at the GI-Seminar *Zeichnen von Graphen*, April 6-9, Schloß Dagstuhl, Germany.

7.1 Graph Drawing using Planarization

While a lot of software exists for hierarchical methods, there are almost no implementations for the method using planarization, although in many applications this method leads to the most pleasant drawings. The reason for this is that a lot of knowledge about planarity testing, embedding and planarization is needed and many complicated data structures and algorithms need to be implemented. The strengths of some members of our group lie exactly in these areas. Therefore, our research mainly focuses on graph drawing using planarization.

There are still many unsolved theoretical and practical questions. The drawing method using planarization transforms a given non-planar graph into a planar graph, and then uses planar graph drawing algorithms to draw the graph.

The idea is to use planar graph theory in order to obtain a good drawing. We use the planarization technique described in [1, 2]. In a first step, the minimum number of edges of G is deleted in order to obtain a planar subgraph. In a second step, a combinatorial embedding of the planar subgraph is determined, that is, the faces are fixed. In a third step, the removed edges are reinserted into our combinatorial embedding so that the number of crossings is minimized. Then,

the crossings are substituted by artificial vertices and the planarized graph can be drawn using any planar graph drawing method.

The maximum planar subgraph problem arising in the first step is an NP-hard combinatorial optimization problem. In earlier work, we have shown that we can efficiently solve practical instances of this problem for graphs of moderate size (up to 80 vertices) to provable optimality [3]. The task in step 2 can be solved in linear time (see, *e.g.*, [4]). However, the number of crossings achieved in step 3 highly depends on the embedding chosen in step 2. Therefore, we are investigating the problem of optimizing over all embeddings in a planar graph (see Section 7.1.1). Here, we got very encouraging and surprising results. The constrained crossing minimization problem arising in step 3 is also an NP-hard problem. We have some hope that our research there will lead to practically efficient algorithms for solving small instances of the general crossing minimization problem in the future. Our current research concerning this problem is described in Section 7.1.2. Section 7.1.3 describes our work done in planar graph drawing algorithms. Some of these algorithms need a compaction phase. The compaction problem arising in graph drawing is similar to the one in VLSI-design but not the same. Recently, we have made significant progress with a new graph-theoretical formulation of the two-dimensional compaction problem (see Section 7.1.4). Some graph drawing algorithms only work for biconnected planar graphs (see, *e.g.*, the ones described in Section 7.1.3). The (NP-hard) planar augmentation problem is to find the minimum number of edges to be added to a given planar graph so that the resulting graph is biconnected and planar. Here, we have given various new polynomial time approximation algorithms (see Section 7.1.5).

References

[1] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing.* Prentice Hall, 1999.

- [2] P. Eades and P. Mutzel. Graph drawing algorithms. In M. Atallah, editor, *CRC Handbook of Algorithms and Theory of Computation*, chapter 9, pages 9–1–9–26. CRC Press, 1999.

[3] M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica, Special Issue on Graph Drawing*, 16(1):33–59, 1996.

[4] K. Mehlhorn and P. Mutzel. On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm. *Algorithmica*, 16(2):233–242, 1996.

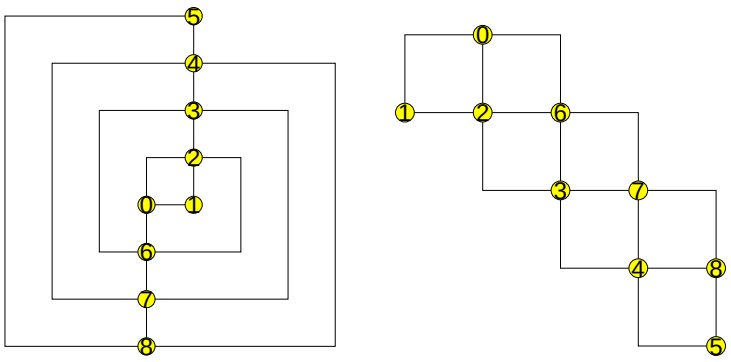
7.1.1 Optimizing over all Embeddings in a Planar Graph

Investigators: Petra Mutzel and René Weiskircher

A *combinatorial embedding* of a planar graph is an equivalence class of planar drawings of the graph. It is defined by the circular sequence of the incident edges around each node or, alternatively, by the list of cycles in the graph that bound faces in the drawing.

In the field of graph drawing, there are some interesting problems that are polynomial-time solvable when the embedding of the graph in question is part of the input but are NP-hard when the embedding is part of the solution.

Bend Minimization Among these problems is the *bend minimization problem* for orthogonal drawings. In an orthogonal drawing of a graph, all edges consist only of horizontal and vertical segments. When we draw each vertex as a point, we can only attach at most four edges. Tamassia has shown that the bend-minimization problem for planar graphs with maximum degree four is solvable in polynomial time if the embedding is fixed [7]. Figure 7.1 shows two bend minimum drawings for the same graph when the embedding is fixed and free, respectively. Often, the quality



(a) (b)

Figure 7.1: Bend minimum drawings for (a) a fixed and (b) a free embedding.

of a drawing improves if the number of bends is minimized over the set of all embeddings. Garg and Tamassia have shown that the problem becomes NP-hard, when the embedding is not part of the input [3].

Crossing Minimization We consider the following problem: We are given a planar graph and an additional edge that is not contained in the graph. The task is to produce a drawing of the graph plus the additional edge with as few crossings as possible. When the embedding is fixed, this problem can be solved efficiently by solving essentially a shortest path problem on the dual graph of the original graph. However, the number of crossings obtained highly depends on the chosen embedding. When the embedding is not part of the problem, we conjecture that the problem becomes NP-hard.

Application of Integer Linear Programming We first considered the problem of optimizing over all embeddings in a planar graph where the objective function is given by some linear combination of all cycles in the graph. Our aim was to formulate an integer linear program (ILP) whose set of feasible solutions corresponds to the set of all combinatorial embeddings.

Our first idea was to use the characterization of planar graphs given in [4]. MacLane proves that a 2-connected graph is planar if and only if there is a basis of its cycle space where each edge is contained in at most two cycles of the basis. When we formulate this as an ILP, we need one variable for each cycle of the graph and we need an exponential number of constraints to guarantee that the set of cycles in every solution of the ILP is a basis of the cycle space.

When we realized that this approach cannot be used for an implementation because of the size of the ILP, we developed a different ILP, that is constructed recursively using a data structure called SPQR-tree developed by Di Battista and Tamassia [1]. SPQR-trees represent the decomposition of a biconnected planar graph into triconnected components and they can be used to enumerate all combinatorial embeddings of a biconnected planar graph. Di Battista *et al.* use this fact in a branch and bound algorithm for bend minimization over all combinatorial embeddings [2]. However, their algorithm for minimizing the number of bends in an orthogonal drawing can take one hour on a graph with 60 vertices. We hope that a branch-and-cut approach using our ILP will be faster and therefore able to handle larger graphs.

Our algorithm works by choosing a node of the SPQR-tree, splitting the tree at this node into smaller SPQR-trees and then recursively computing the ILPs for the smaller trees. Then we construct the ILP for the original problem by merging the ILPs we have constructed using the SPQR-trees generated by the splitting operation. The splitting process stops when the SPQR-trees have only one inner node. The ILPs for these trees are explicitly defined in our algorithm. Our recursive construction guarantees that we only get variables for those cycles in the graph that appear as face cycles in at least one combinatorial embedding.

A first implementation has shown surprising results: In our experiments on a benchmark set of graphs and on randomly generated graphs with an extremely high number of embeddings, the number of variables and constraints in the ILPs grow linearly with the size of the graphs (maximal degree four). The time needed to construct the ILPs was sub-exponential. For example, the ILP for a graph with 500 vertices and 10^{19} combinatorial embeddings contained only about 1000 variables and 2500 constraints. The typical sizes of ILPs for real-world graphs on 100 vertices are about 100 variables and 250 constraints. The time for constructing the ILPs was about 5 seconds for real-world instances and 5 minutes for the hard random instances. However, the ILPs could be solved within 0.06 seconds and 2 seconds, respectively, for various objective functions [6, 5].

So we are hopeful that branch-and-cut algorithms for solving the bend and crossing minimization problems mentioned above will be able to solve problem instances of reasonable sizes in acceptable running time.

References

[1] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997, October 1996.

[2] P. Bertolazzi, G. Di Battista, and W. Didimo. Computing orthogonal drawings with the minimum number of bends. *Lecture Notes in Computer Science*, 1272:331–344, 1998.

[3] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *Lecture Notes in Computer Science*, 894:286–297, 1995.

[4] S. MacLane. A combinatorial condition for planar graphs. *Fundamenta Mathematicae*, 28:22–32, 1937.

- [5] P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph. Technical report, Max-Planck-Institut für Informatik, Saarbrücken, 1998.

- [6] P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

[7] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.

7.1.2 Constrained Crossing Minimization

Investigators: Petra Mutzel and Thomas Ziegler

For the edge re-insertion step in the planarization method, we investigate the *constrained crossing minimization problem* defined as follows. We are given a connected, planar graph $G = (V, E)$, a combinatorial embedding $\Pi(G)$ of G, and a set of pairwise distinct edges $F \subseteq V \times V$, and we want to find a drawing of $G' = (V, E \cup F)$ such that the combinatorial embedding $\Pi(G)$ of G is preserved and the number of edge crossings is minimized.

In practice this problem has been attacked so far using the following observation: For inserting one edge, the problem can be solved optimally in polynomial time by computing a shortest path in the extended dual graph (*i.e.*, the dual graph extended by some vertices and edges). The common heuristics iteratively insert the edges using this dual graph approach. However, the result is not always acceptable. We think that an optimal solution of the problem leads to much nicer drawings.

Our investigations have shown that the constrained crossing minimization problem is NP–hard. For this, we have used the FIXED LINEAR CROSSING NUMBER PROBLEM, which was proven to be NP–complete in [3].

In our approach for attacking practical instances of the constrained crossing minimization problem we use the extended dual graph D of G. Our first idea was to search for a set of paths in the extended dual graph, such that the sum of the lengths of the paths plus the number of crossings between the paths is minimum. However, it turned out that paths in the extended dual graph are not sufficient to formulate the constrained crossing minimization problem correctly.

Figure 7.2 shows an example for which the path formulation would not give the optimum solution of the constrained crossing minimization problem. Consider the graph induced by the solid edges. We want to add the four dashed edges. When we restrict our attention to paths in the extended dual graph, we would get a solution of three crossings while the optimum solution has only two crossings.

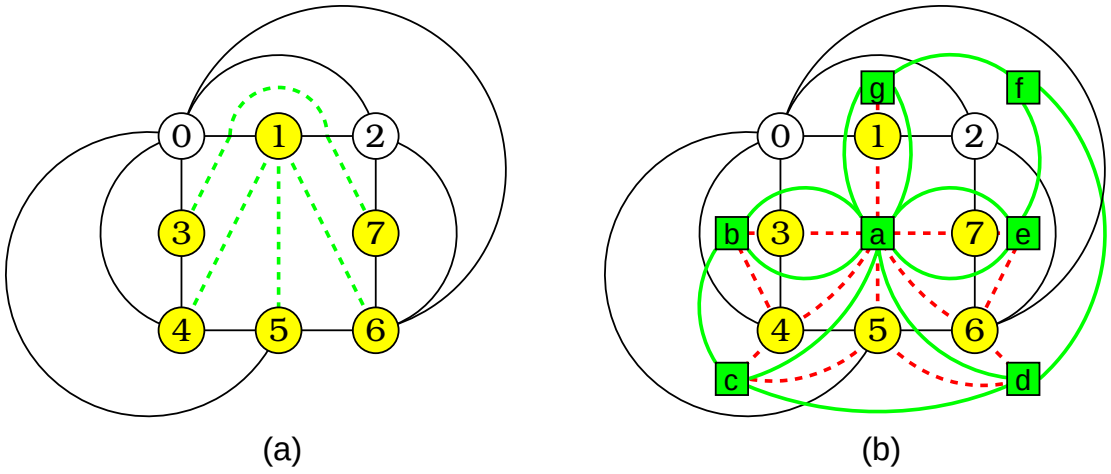


Figure 7.2: (a) An example for which the path formulation is not sufficient; (b) the dual graph is shown in grey (solid lines) extended by the dashed lines shown in dark grey.

Hence, we considered walks instead of paths in the extended dual graph. The only difference between a walk and a path is, that a walk can use vertices and edges of the graph more than once.

We have defined the *shortest crossing walks problem* as follows. Given a connected, planar graph $D = (V_D, E_D)$ with a combinatorial embedding $\Pi(D)$ and a set $F_D \subseteq V_D \times V_D$ of distinct pairs of vertices, called *commodities*, find a set of walks in D such that there is exactly one walk for each commodity, no walk uses a vertex that appears in F_D as an inner vertex, and the sum of the lengths of the walks plus the number of crossings between walks is minimum.

In [5] we have shown that the constrained crossing minimization problem is polynomially equivalent to the shortest crossing walks problem on the extended dual graph. The shortest crossing walks problem is of combinatorial rather than geometric nature.

We have found an integer linear programming formulation for the shortest crossing walks problem, where we use binary variables for pairs of adjacent edges to describe the corresponding walks for every commodity $k \in F_D$ and a binary variable z_v^{kl} for every vertex $v \in V_D$ and every pair of commodities $k, l \in F_D$ to describe crossings between walks. Our objective is to minimize the sum of the lengths of the walks plus the number of crossings between the walks. We used this formulation to derive a branch-and-cut algorithm for the constrained crossing minimization problem. We basically start with the constraints describing the walks for the commodities and use the constraints describing crossings between walks as cutting planes. Moreover, we use additional valid inequalities for the shortest crossing walks problem in our algorithm.

At the moment we are implementing this algorithm using LEDA and ABACUS [4, 2]. We will do computational experiments on the benchmark set of graphs from the University of Rome III [1]. We expect that our results will improve the results of the best known heuristics considerably and that the additional valid inequalities strengthen the formulation of the problem significantly.

Since LP-formulations coding a set of paths are quite common in the mathematical programming community, we decided to implement a branch-and-cut algorithm based on the path formulation, too. This code has already been tested on the benchmark set. It turned out that this approach already improves the results of the best known heuristics.

References

[1] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7:303–326, 1997.

[2] M. Jünger and S. Thienel. Introduction to ABACUS - A Branch-And-CUt System. *Technical Report No. 97.263, Institut für Informatik, Universität zu Köln*, 1997. To appear in Operations Research Letters, 1999.

[3] S. Masuda, K. Nakajima, T. Kashiwabara, and T. Fujisawa. Crossing minimization in linear embeddings of graphs. *IEEE Transactions on Computers*, 39(1):124–127, 1990.

[4] K. Mehlhorn and S. Näher. LEDA: A platform for combinatorial and geometric computing. *Comm. Assoc. Comput. Mach.*, 38:96–102, 1995.

- [5] P. Mutzel and T. Ziegler. The constrained crossing minimization problem – a first approach. In P. Kall and H.-J. Lüthi, editors, *Operations Research Proceedings 1998*, pages 125–134. Springer-Verlag, 1999.

7.1.3 Planar Graph Drawing Algorithms

Investigators: Carsten Gutwenger, Gunnar Klau, and Petra Mutzel

In our last report, we described our research on straightline planar drawing methods, on the mixed model method, and on the quasi-orthogonal drawing method. This work is still ongoing; we have improved some of the methods. In addition, we have extended our research to pure orthogonal drawings. But so far no papers have been published on that. Here, we only mention our improvement on the mixed model algorithm.

In [4], we have presented a linear time algorithm that constructs a planar polyline grid drawing of any plane graph with n vertices and maximum degree d on a $(2n-5) \times (\frac{3}{2}n - \frac{7}{2})$ grid with at most $5n - 15$ bends and minimum angle $> \frac{2}{d}$. In the constructed drawings, every edge has at most three bends and length $O(n)$. To the best of our knowledge, this algorithm achieves the best simultaneous bounds concerning the grid size, angular resolution, and number of bends for planar grid drawings of high-degree planar graphs. Besides the nice theoretical features, the practical drawings are aesthetically very pleasing, since they are almost orthogonal; the non-orthogonal lines

are short, and the minimal angles are quite big. An implementation of our algorithm is available with the AGD-Library [2, 1].

Our algorithm is based on ideas by Kant for polyline grid drawings for triconnected plane graphs [5]. In particular, our algorithm significantly improves upon his bounds on the angular resolution and the grid size for non-triconnected plane graphs. In this case, Kant could show an angular resolution of $\frac{4}{3d+7}$ and a grid size of $(2n-5) \times (3n-6)$, only.

The algorithm proceeds similar to the straight-line algorithms suggested first by De Fraysseix, Pach, and Pollack [3]. In a first step, the vertices are ordered according to a certain canonical ordering, and in the second step, the vertex boxes are placed incrementally at certain grid points. Roughly speaking, a vertex box consists of the vertex plus the first parts of the incident edges, that are regularly distributed around the vertex.

Our improvement comes from a generalization of Kant's canonical ordering for triconnected planar graphs to biconnected planar graphs and from improvements in the placement step.

References

- [1] AGD. *AGD User Manual.* Max-Planck-Institut Saarbrücken, Universität Halle, Universität Köln, 1998. Available via `http://www.mpi-sb.mpg.de/AGD/`.
- [2] D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD-Library: A library of algorithms for graph drawing. In G. F. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE '97)*, 1997. Venice, Italy, Sept. 11-13, http://www.dsi.unive.it/∼wae97/.

[3] H. De Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

- [4] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In S. H. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 167–182. Springer-Verlag, 1998.

[5] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica, Special Issue on Graph Drawing*, 16(1):4–32, 1996.

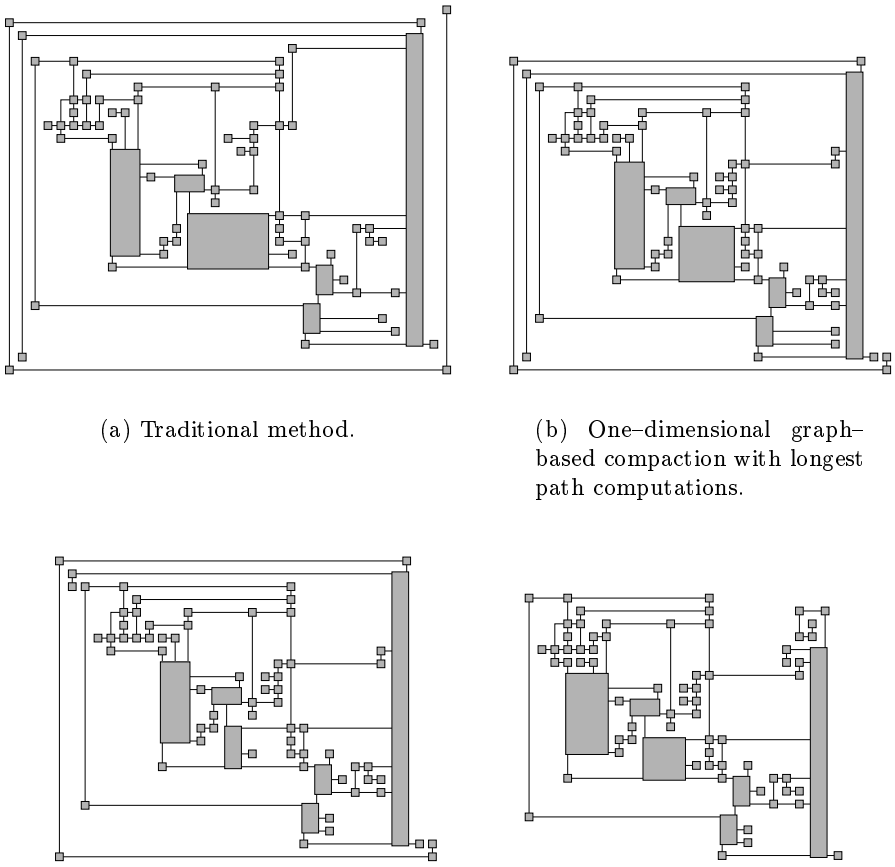
7.1.4 Compacting Graph Drawings

Investigators: Gunnar Klau and Petra Mutzel

Orthogonal graph drawings are highly accepted in practice because of their excellent readability. In many application areas (*e.g.*, database design, software engineering, and VLSI layout) orthogonality is a strict requirement for the output of the layout algorithms.

Many orthogonal graph drawing algorithms — especially the ones within the topology–shape–metrics paradigm [2] — suffer from the absence of compaction algorithms. In orthogonal grid embeddings produced by the traditional compaction method in [13] many edges are drawn too long and force the layout to use a large amount of drawing space (see Fig. 7.3(a)). One–dimensional compaction strategies known from VLSI design may help in many cases but there are a lot of instances for which both directions are blocked and no further one–dimensional improvement is possible (see Figs. 7.3(b) and 7.3(c)).

We have concentrated on the following compaction problem in graph drawing: Given an orthogonal representation H describing the shape of any drawing of the input graph, produce an orthogonal grid embedding with given shape and minimum total edge length. We have developed a compaction framework [9, 11] to solve this and similar problems to optimality (see Fig. 7.3(d)). The framework is open to all different orthogonal drawing standards:



(a) Traditional method.

(b) One–dimensional graph–based compaction with longest path computations.

(c) One–dimensional graph–based compaction with flow computations.

(d) Optimal compaction method (w.r.t. total edge length).

Figure 7.3: The result of four different compaction algorithms.

- *Pure orthogonal drawings* are only admissible for 4–graphs (no vertex has a higher degree than four). Here, vertices are mapped to grid points and edges to paths in the grid.
- In *orthogonal box drawings* vertices are mapped to rectangles in the grid and edges to paths. This drawing standard can be used when the vertex degree exceeds four. Subclasses of this standard are the *big node model* [7] and the *TSS model* [3]; a related class is the *quasi–orthogonal* model [10].
- *Kandinsky-like drawings* represent vertices as points in a coarse grid and edges as paths in a finer grid. The model has been introduced in [6].

In our new approach we characterize feasible solutions of the compaction problem in terms of extensions of the so–called *constraint graphs* in x– and y–direction. We define *segments* as maximally connected chains of horizontal or vertical edges forming the nodes in the constraint graphs. Unlike in VLSI approaches, we only introduce arcs in the constraint graphs if the relative position between two elements in the compaction process is already coded in the given orthogonal

representation and thus known in advance. We call the pair of graphs *complete* if and only if the graphs are acyclic and each pair of segments is separated according to one of four rules.

We reformulate the two–dimensional compaction problem as the search for a *complete extension* of the given pair of constraint graphs. Among all extensions with this property we look for the one leading to the drawing with minimum total edge length. We constructively detect those instances having only one possible extension. In these cases we solve the compaction problem in polynomial time.

We formulate the resulting graph–theoretical problem as an integer linear program and present a branch–and–cut algorithm to solve the two–dimensional compaction problem to optimality. The algorithm is based on the libraries ABACUS [8] and LEDA [12] and is realized as a module inside our graph drawing library AGD [1]. Our computational results on a benchmark set of 11,582 graphs [5] have shown that we are able to solve the two–dimensional compaction problem for all the instances in short computation time: For 95% of the instances it took less than one second and for 99% less than five seconds to compute the optimal solution. Furthermore, the experiments have shown that often it is worthwhile to look for the optimally compacted drawing. The total edge lengths have been improved up to 37.0% and 65.4%, respectively, as compared to iterated one–dimensional compaction and the traditional method.

Recently, Bridgeman *et al.* have presented an independent study of polynomial–time compactable orthogonal representations [4]. They devise a class of so–called *turn–regular* representations and give a linear time algorithm to find optimal drawings for these instances. The class, however, is a subclass of representations for which our compaction framework works in polynomial time.

Our framework is open to a broad variety of related problems: In addition to the capability to cope with the different drawing standards it can handle different versions of the compaction problem, *e.g.*, insertion and removal of bends, detecting unnecessary crossings and many more. At the moment, we are formulating the solution of a *graph labeling problem* (combined compaction and labeling) with our framework.

References

- [1] AGD. *AGD User Manual.* Max-Planck-Institut Saarbrücken, Universität Halle, Universität Köln, 1998. Available via `http://www.mpi-sb.mpg.de/AGD/`.

[2] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing.* Prentice Hall, 1999.

[3] T. Biedl, B. Madden, and I. Tollis. The three–phase method: A unified approach to orthogonal graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 391–402. Springer-Verlag, 1997.

[4] S. Bridgeman, G. Di Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara. Turn–regularity and optimal area drawings of orthogonal representations. Technical report, Dipartimento di Informatica e Automazione, Università degli Studi di Roma Tre, 1999. To appear.

[5] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7:303–326, 1997.

[6] U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 254–266. Springer-Verlag, 1996.

[7] U. Fößmeier and M. Kaufmann. Algorithms and area bounds for nonplanar orthogonal drawings. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 134–145. Springer-Verlag, 1997.

[8] M. Jünger and S. Thienel. Introduction to ABACUS - A Branch-And-CUt System. *Technical Report No. 97.263, Institut für Informatik, Universität zu Köln*, 1997. To appear in Operations Research Letters, 1999.

- [9] G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. Technical Report MPI–I–98–1–031, Max–Planck–Institut für Informatik, Saarbrücken, December 1998.

[10] G. W. Klau and P. Mutzel. Quasi–orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max–Planck–Institut für Informatik, Saarbrücken, 1998.

- [11] G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

[12] K. Mehlhorn and S. Näher. LEDA: A platform for combinatorial and geometric computing. *Comm. Assoc. Comput. Mach.*, 38:96–102, 1995.

[13] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.

7.1.5 Planar Augmentation

Investigators: Petra Mutzel

Many planar graph drawing algorithms need as input a biconnected planar graph. In order to use those algorithms for non-biconnected graphs, the planar augmentation problem needs to be solved. Given a planar graph, the planar augmentation problem is to add the minimum number of edges such that the resulting graph is still planar and biconnected. The problem was introduced by Kant and Bodlaender [7]. They also showed the NP-hardness of this problem and suggested a polynomial time algorithm that they claimed approximates the optimum solution within a factor of $\frac{3}{2}$. However, this algorithm is not correct [3, 6]. If all the mistakes are corrected, then it is instead a 2-approximative algorithm (see [3]). In [3], we have given a polynomial time $\frac{5}{3}$-approximation algorithm for the planar augmentation problem with running time $O(n^2T)$, where T is the amortized time bound per insertion operation in incremental planarity testing. The quality of our algorithm improves upon the quality of the previously known approximation algorithm. The factor of $\frac{5}{3}$ is tight. We have implemented our approximation algorithm and have been able to evaluate its behaviour in practice.

In [4, 8], we have introduced a branch-and-cut method for the planar augmentation problem using the polyhedral structure of the associated polytope (see last report). In [2], we have designed a new branch-and-cut method that has been implemented using the system ABACUS [5]. In the last report, we reported on our encouraging computational results: instances of up to 200 vertices can be solved to optimality within a few seconds. Moreover, all the practical instances arising from the benchmark set used in [1] can be solved within a few seconds. This is the first time that any instances of these sizes can be solved to optimality at all.

The fact that we have been able to solve all the benchmark instances to optimality gives us the opportunity to compare the solutions of our new approximation algorithm with optimum solutions. Surprisingly, our approximation algorithm has given the optimum solutions in many cases. And in almost all cases the solution given by the algorithm contained only one more edge (or in rare cases two or three more edges) than the optimum solution.

Very recently, we could improve the approximation factor from $\frac{5}{3}$ to $\frac{3}{2}$ using a new linear time algorithm for the planar augmentation problem in the case where the combinatorial embedding has been fixed [9]. So far, no paper has been published. The idea of the new algorithm is to use an

algorithm similar to the $\frac{5}{3}$-approximation algorithm in a first phase, then compute a combinatorial embedding of the resulting graph, and in the third phase delete the augmented edges again and solve the fixed planar augmentation problem. For a description of the algorithm in more detail, see [9, 10].

References

[1] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of three graph drawing algorithms. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 306–315, 1995.

[2] S. Fialko. Das Planare Augmentierungsproblem. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.

- [3] S. Fialko and P. Mutzel. A new approximation algorithm for the planar augmentation problem. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pages 260–269, San Francisco, California, 1998. ACM Press.

[4] M. Jünger and P. Mutzel. The polyhedral approach to the maximum planar subgraph problem: New chances for related problems. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 119–130. Springer-Verlag, 1995.

[5] M. Jünger and S. Thienel. Introduction to ABACUS - A Branch-And-CUt System. *Technical Report No. 97.263, Institut für Informatik, Universität zu Köln*, 1997. To appear in Operations Research Letters, 1999.

[6] G. Kant, 1997. Personal communication.

[7] G. Kant and H. L. Bodlaender. Planar graph augmentation problems. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pages 286–298. Springer-Verlag, 1991.

[8] P. Mutzel. A polyhedral approach to planar augmentation and related problems. In P. Spirakis, editor, *Algorithms – ESA '95, Third Annual European Symposium*, volume 979 of *Lecture Notes in Computer Science*, pages 494–507. Springer-Verlag, 1995.

- [9] P. Mutzel. A $\frac{3}{2}$-approximation algorithm for the planar augmentation problem. Extended Abstract, 1998.

- [10] P. Mutzel. New algorithms for the planar augmentation problem. Slides available via `http://www.mpi-sb.mpg.de/~wae98/`, 1998. Invited talk held at the 2nd Workshop on Algorithm Engineering, WAE '98, Saarbrücken.

7.2 Hierarchical Graph Drawing

Investigators: Petra Mutzel

Hierarchical graph drawing methods generate drawings that represent the hierarchy in the given data. Amongst the most commonly used hierarchical drawing methods are the layer-based methods.

If the graph is acyclic, then in a first step, the vertex set is partitioned into subsets V_i, such that all arcs are directed from V_i to V_j with $i \leq j$. So, the vertices in each subset V_i can be placed on a horizontal line such that all arcs point in one direction, say downward. In a second step of the algorithm, the vertices in each layer V_i are permuted such that the overall number of crossings, when the arcs are drawn as straight-lines, should be minimized.

In Section 7.2.1 we focuse on the second step, namely, the k-layer crossing minimization problem. In the last report, we described our research on the k-layer planarization problem. This is an alternative approach to k-layer crossing minimization. Our new research there has mainly been

concerned with k-layer planarity testing algorithms (see Section 7.2.2). In some applications, a bipartite, planar graph G is given that should be drawn without crossings such that the partition is clearly visible. See Section 7.2.3 for our work done in this direction.

7.2.1 K-Layer Crossing Minimization

Investigator: Petra Mutzel

Recently, we have started to study the k-layer crossing minimization problem [2].

Let us first address the case where $k = 2$. Here, the objective is to find the minimum number of edge-crossings among the two shores of a bipartite graph, when both shores can be permuted. We have formulated this problem as an integer linear program, in which the optimal solution corresponds to a minimum crossing solution. Then, we have extended this integer programming formulation to a more general case – the proper *Multi Layer Crossing Minimization* problem – which is to determine the minimum number of crossings in a (proper) multi-layered graph.

Using polyhedral combinatorics, we have investigated the structure of the polytope associated with the 2-layer crossing minimization problem. We have found several classes of facet-defining inequalities for this polytope. Most of the inequalities have been derived from a combinatorial characterization of 2-planar graphs, *i.e.*, graphs which can be drawn on two layers without edge-crossings [1, 4]. Most of these inequalities are still valid and useful for the k-layer crossing minimization problem.

In order to use these inequalities as cutting planes in a branch-and-cut algorithm, we have investigated separation procedures for each class of facet-defining inequalities. Preliminary numerical experiments with a simple cutting-plane algorithm have shown that the corresponding cuts indeed strengthen the relaxation. However, a lot of effort is still needed to get an efficient algorithm that will be able to solve the k-layer crossing minimization problem for moderately sized instances that occur in graph drawing. So far, for two layers our previously developed branch-and-bound algorithm, which is based on a branch-and-cut algorithm for the one-layer fixed crossing minimization problem, is still superior to the new cutting plane algorithm [3].

References

[1] F. Harary and A. Schwenk. A new crossing number for bipartite graphs. *Utilitas Mathematica*, 1:203–209, 1972.

- [2] M. Jünger, E. Lee, P. Mutzel, and T. Odenthal. A polyhedral approach to the multi-layer crossing number problem. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 13–24. Springer-Verlag, 1997.

[3] M. Jünger and P. Mutzel. Exact and heuristic algorithms for 2-layer straightline crossing minimization. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996. Revised version published in Journal of Graph Algorithms and Applications (JGAA).

[4] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 318–333. Springer-Verlag, 1997.

7.2.2 K-Layer Planarity Testing

Investigator: Petra Mutzel

In the last report, we described our ideas for an alternative method to k-layer crossing minimization called the k-layer planarization problem. The k-layer planarization problem asks for removing the minimal number of edges such that the resulting graph is k-layer planar. In the final diagram the removed edges are reinserted as straight lines into the k-layer planar drawing. Our hope was that this problem might be easier to attack than the k-layer crossing minimization problem. Last time, we reported on our work for $k = 2$ [10, 11]. In order to investigate the k-layer planarization problem for $k > 2$, we first needed to have a closer look at k-layer planarity testing algorithms.

Heath and Pemmaraju [2] have suggested a linear time k-layer planarity testing algorithm. However, we were able to show in [5] that their algorithm, based on the data structure of PQ-trees, is not correct.

This is not the first time that we have discovered non-correctness of published algorithms based on PQ-trees. Our experience shows that the authors make implicit assumptions about certain properties of PQ-trees that are not correct. We have also shown [7, 5] problems in the articles by Jayakumar *et al.* [3] and Kant [8] on maximal planarization algorithms due to the same problem. Our paper [5] indicates why we believe that the chance for solving the maximal planar subgraph problem in polynomial time using PQ-trees is small.

However, for k-layer planarity testing we could give a correct linear time algorithm [6]. Our algorithm generalizes the algorithm by Nardelli and Di Battista, which only works for the special case of one sink or one source [1]. An implementation has shown that the algorithm is extremely fast.

Very recently, Jünger and Leipert have given a linear time embedding algorithm based on the k-layer planarity testing algorithm [4, 9]. This gives us the basis to continue our research on the k-level planarization problem in the future.

References

[1] G. Di Battista and E. Nardelli. An algorithm for testing planarity of hierarchical graphs. In G. Tinhofer and G. Schmidt, editors, *Graph Theoretic Concepts in Computer Science: Proc. Internat. Workshop WG '86 (Bernierd June 1986)*, volume 246 of *Lecture Notes in Computer Science*, pages 277–289, Berlin, 1987. Springer-Verlag.

[2] L. S. Heath and S. V. Pemmaraju. Recognizing leveled-planar dags in linear time. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 300–311. Springer-Verlag, 1996.

[3] R. Jayakumar, K. Thulasiraman, and M. N. S. Swamy. $O(n^2)$ algorithms for graph planarization. *IEEE Trans. on Computer-Aided Design*, 8:257–267, 1989.

[4] M. Jünger and S. Leipert. On embedding k-level planar graphs. Manuscript, 1999.

- [5] M. Jünger, S. Leipert, and P. Mutzel. Pitfalls of using PQ-trees in automatic graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204. Springer-Verlag, 1997.

- [6] M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In S. H. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer-Verlag, 1998.

- [7] M. Jünger, S. Leipert, and P. Mutzel. A note on computing a maximal planar subgraph using PQ-trees. *IEEE Transactions on Computer-Aided Design*, 17(7), 1998.

[8] G. Kant. An $o(n^2)$ maximal planarization algorithm based on PQ-trees. Technical Report RUU-CS-92-03, Dept. of Computer Science, Utrecht University, 1992.

[9] S. Leipert. *Level Planarity Testing and Embedding in Linear Time.* PhD thesis, Institut für Informatik, Universität zu Köln, 1998.

[10] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 318–333. Springer-Verlag, 1997.

- [11] P. Mutzel and R. Weiskircher. Two-layer planarization in graph drawing. In K.-Y. Chwa and O. H. Ibarra, editors, *Algorithms and Computation, 9th International Symposium, ISAAC '98*, volume 1533 of *Lecture Notes in Computer Science*, pages 69–78, Taejon, Korea, 1998. Springer-Verlag.

7.2.3 Drawing Planar Partitions

Investigator: Petra Mutzel

Assume that $G = (V, E)$ is a graph and $V = A \cup B$ is a partition of the vertices of G. How should we draw G such that the partition is clearly visible? Our study of this question was motivated by a competition graph (Graph B) of the Graph Drawing Competition 1996 (see [3]), which is a graph of telephone calls and turned out to be bipartite, hence had a natural partition structure. A drawing of G such that the partition is clearly visible aids in understanding the structure of these problems better.

In [2], we consider *straight-line* and *poly-line grid drawings*, that is, vertices are drawn as points on a grid (*i.e.*, with integer coordinates), and edges are drawn either as straight lines or as sequences of straight-line segments where the bend points lie on a grid as well. We study only *planar partitions*, *i.e.* the graph $G = (A \cup B, E)$ is planar, and we require that the drawing of G has no crossing. We study planar drawings of G in which the vertex classes A and B are separated by a horizontal line (so-called HH-drawings).

Not every planar partition has a planar HH-drawing if the edges are required to be drawn y-monotone, *i.e.*, with monotonically increasing y-coordinates. In [2], we provide necessary and sufficient conditions for the existence of planar y-monotone HH-drawings. In [1], it is shown that these conditions can be tested in linear time. One surprising corollary of our characterization is that every bipartite planar graph has a planar y-monotone HH-drawing.

The proof of sufficiency yields an algorithm for planar y-monotone HH-drawings with area $O(|V|^2)$ and at most one bend per edge. We prove that straight-line HH-drawings of polynomial area are not always possible, viz., there exists a graph class for which any planar straight-line HH-drawing requires exponential area. Finally, we drop the monotonicity-requirement, and prove that then every planar partition has a planar HH-drawing with at most three bends per edge.

References

[1] T. Biedl. Drawing planar partitions III: Two constrained embeddings. Technical Report RRR1398, RUTCOR, Rutgers University, 1998.

- [2] T. Biedl, M. Kaufmann, and P. Mutzel. Drawing planar partitions II: HH-drawings. In J. Hromkovic and O. Sykora, editors, *Proceedings of the 24-th Workshop on Graph-Theoretic Concepts in Computer Science (WG '98)*, volume 1517 of *Lecture Notes in Computer Science*, pages 124–136. Springer-Verlag, 1998.

[3] P. Eades, J. Marks, and S. North. Graph drawing contest report. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 129–138. Springer-Verlag, 1997.

7.3 Applications, Practical Projects and Software

7.3.1 Graph Drawing Applications

Investigator: Rudolf Fleischer

The practical applicability of graph drawing theory was the focus of a recent study [2], where we analyzed 370 graph drawing papers and asked some real users in need of graph drawing tools about their experience. We have discovered several problems that cannot be solved satisfactory so far using the current methods of graph drawing. For example, the graphs arising in some applications have a strong semantic background which is not captured by the existing graph drawing tools. However, research in the field of graph drawing is still young.

7.3.2 Software and Practical Projects

Investigators: Ralf Brockenauer, Carsten Gutwenger, Christoph Hundack, Gunnar Klau, Petra Mutzel, René Weiskircher, and Thomas Ziegler

Great progress on applications and practical graph drawing tools has been made during the last few years. Since this trend is not only present in our group but also in other graph drawing groups, we feel encouraged to go on. Our experience has shown that from practice often new interesting theoretical problems arise (for example, the planar augmentation problem, embedding problems, the two-dimensional graph drawing compaction problem or high-degree bend minimization).

Most of our theoretical work and algorithms has been transferred into software. In general, this software becomes part of AGD, our library of **A**lgorithms for **G**raph **D**rawing (see, *e.g.*, [1]). AGD is described in Section 8.2.

Moreover, we have further developed the program *ArchEd*, which is a tool for archaeologists [3] (for a description see, *e.g.*, the last report). *ArchEd* is publically available on the web via `http://www.mpi-sb.mpg.de/~arche`. We get lots of feedback from archaeologists all over the world.

We are getting industrial funds for cooperating on a project involving drawings of finite state machines that describe the control of computer integrated manufacturing processes. Moreover, we got a grant from the German Ministry of Education, Science, and Technology for which we are investigating the labeling problem in connection with graph drawing. Our grant from the German Science Foundation is mostly for developing AGD further. Very recently, we got a EU-grant for developing a documentation tool for electronic questionnaires. For a detailed description of our projects and grants, see Section 14.

References

- [1] D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD-Library: A library of algorithms for graph drawing. In G. F. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE '97)*, 1997. Venice, Italy, Sept. 11-13, http://www.dsi.unive.it/∼wae97/.
- [2] R. Fleischer and C. Hirsch. Applications of graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation. Presented at the GI-Seminar *Zeichnen von Graphen*, April 6-9, Schloß Dagstuhl, Germany.
- [3] C. Hundack, P. Mutzel, I. Pouchkarev, and S. Thome. *ArchE*: A graph drawing system for archaeology. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 297–302. Springer-Verlag, 1997.

8 Software Libraries

Software is the physical infrastructure of the information age and fundamental to economic success as well as scientific and technical research [1]. Software libraries are important for the efficient development of robust, reliable software. Furthermore, they play a key role in the technology transfer from academia to industry. For these and other reasons, the US President's Information technology Advisory Committee [1] recently recommended sponsoring libraries of software components in subject area domains.

In our group, we are well ahead on this road. We have continued our work on the LEDA platform of combinatorial and geometric computing, which is developed in cooperation with the group of Stefan Näher in Halle, and on the computational geometry algorithms library CGAL, which is developed in cooperation with several research institutes and universities in Europe and Israel. AGD, a library of algorithms for graph drawing, is developed in cooperation with Stefan Näher's group in Halle and Michael Jünger's group in Köln. A new software library has been added: BALL, a Biochemical Algorithms Library. The functionality of LEDA has been expanded in several LEDA extension packages, partially by contributors outside of the institute.

References

[1] President's Information Technology Advisory Committee. Research focal points and societal implications, 1998. `http://www.ccic.gov/ac/interim/section_3.html`.

8.1 LEDA

LEDA (Library of Efficient Data Types and Algorithms) is a C++ library for combinatorial and geometric computing. It is developed in joint work with Stefan Näher's group in Halle.

8.1.1 The LEDA Book

Investigator: Kurt Mehlhorn

Stefan Näher and Kurt Mehlhorn finally finished the LEDA-book in April of 1999. The book will appear with Cambridge University Press in the fall of 1999. The book is about 1000 pages in length and covers all aspects of the LEDA system. We quote from the preface of the book.

> LEDA (Library of Efficient Data Types and Algorithms) is a C++ library of combinatorial and geometric data types and algorithms. It offers
>
> **Data Types,** such as random sources, stacks, queues, maps, lists, sets, partitions, dictionaries, sorted sequences, point sets, interval sets, ... ,
>
> **Number Types,** such as integers, rationals, bigfloats, algebraic numbers, and linear algebra,
>
> **Graphs and Supporting Data Structures,** such as node- and edge-arrays, node- and edge-maps, node priority queues and node partitions, iteration statements for nodes and edges, ... ,
>
> **Graph Algorithms,** such as shortest paths, spanning trees, flows, matchings, components, planarity, planar embedding, ... ,
>
> **Geometric Objects,** such as points, lines, segments, rays, planes, circles, polygons, ... ,
>
> **Geometric Algorithms,** such as convex hulls, triangulations, Delaunay diagrams, Voronoi diagrams, segment intersection, ... , and
>
> **Graphical Input and Output.**

The modules just mentioned cover a considerable part of combinatorial and geometric computing as treated in courses and textbooks on data structures and algorithms [1, 4, 2, 3, 5, 6, 10, 8, 11, 12, 13, 14, 16, 17, 18].

From a user's point of view, LEDA is a platform for combinatorial and geometric computing. It provides *algorithmic intelligence* for a wide range of applications. It eases a programmer's life by providing powerful and easy-to-use data types and algorithms which can be used as building blocks in larger programs. It has been used in such diverse areas as code optimization, VLSI design, robot motion planning, traffic scheduling, machine learning and computational biology. The LEDA system is installed at more than 1500 sites.

We started the LEDA project in the fall of 1988. The project grew out of several considerations.

- We had always felt that a significant fraction of the research done in the algorithms area was eminently practical. However, only a small part of it was actually used. We frequently heard from our former students that the intellectual and programming effort needed to implement an advanced data structure or algorithm is too large to be cost-effective. We concluded that *algorithms research must include implementation if the field wants to have maximum impact.*
- We surveyed the amount of code reuse in our own small and tightly connected research group. We found several implementations of the same balanced tree data structure. Thus there was constant reinvention of the wheel even within our own small group.
- Many of our students had implemented algorithms for their master's thesis. Work invested by these students was usually lost after the students graduated. We had no depository for implementations.
- The specifications of advanced data types which we gave in class and which we found in text books, including the one written by one of the authors, were incomplete and not sufficiently abstract to allow to combine implementations easily. They contained phrases of the form: "Given a pointer to a node in the heap its priority can be decreased in constant amortized time". Phrases of this kind imply that a user of a data structure has to know its implementation. As a consequence combining implementations is a non-trivial task. We performed the following experiment. We asked two groups of students to read the chapters on priority queues and shortest path algorithms in a standard text book, respectively, and to implement the part they had read. The two parts would not fit, because the specifications were incomplete and not sufficiently abstract.

We started the LEDA project to overcome these shortcomings by creating a platform for combinatorial and geometric computing. *LEDA should contain the major findings of the algorithms community in a form that makes them directly accessible to non-experts having only limited knowledge of the area.* In this way we hoped to reduce the gap between research and application.

The LEDA system is available from the LEDA web-site at `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

A commercial version of LEDA is available from Algorithmic Solutions Software GmbH at `http://www.algorithmic-solutions.de`.

LEDA can be used with almost any C++ compiler and is available for UNIX and WINDOWS systems. The LEDA mailing list (see the LEDA web page) facilitates the exchange of information between LEDA users.

This book provides a comprehensive treatment of the LEDA system and its use. We treat the architecture of the system, we discuss the functionality of the data types and algorithms available in the system, we discuss the implementation of many modules of the system, and we give many examples for the use of LEDA. We believe that the book is useful to five types of readers: readers with a general interest in combinatorial and geometric computing, casual users of

LEDA, intensive users of LEDA, library designers and software engineers, and students taking an algorithms course.

The book is structured into fourteen chapters.

Chapter 1, Introduction, introduces the reader to the use of LEDA and gives an overview of the system and our design goals.

Chapter 2, Foundations, discusses the basic concepts of the LEDA system. It defines key concepts, such as type, object, variable, value, item, copy, linear order, and running time, and it relates these concepts to C++. We recommend that you read this chapter quickly and come back to it as needed. The detailed knowledge of this chapter is a prerequisite for the intensive use of LEDA. The casual user should be able to satisfy his needs by simply modifying example programs given in the book. The chapter draws upon several sources: object-oriented programming, abstract data types, and efficient algorithms. It lays out many of our major design decisions which we call LEDA axioms.

Chapters 3 to 12 form the bulk of the book. They constitute a guided tour of LEDA. We discuss numbers, basic data types, advanced data types, graphs, graph algorithms, embedded graphs, geometry kernels, geometry algorithms, windows, and graphwins. In each chapter we introduce the functionality of the available data types and algorithms, illustrate their use, and give the implementation of some of them.

Chapter 13, Implementation, discusses the core part of LEDA, *e.g.*, the implementation of parameterized data types, implementation parameters, memory management, and iteration.

Chapter 14, Documentation, discusses the principles underlying the documentation of LEDA and the tools supporting it.

The book can be read without having the LEDA system installed. However, access to the LEDA system will greatly increase the *joy of reading*.

The demo directory of the LEDA system contains numerous programs that allow the reader to exercise the algorithms discussed in the book. The demos give a feeling for the functionality and the efficiency of the algorithms, and in a few cases even animate them.

The book can be read from cover to cover, but we expect few readers to do it. We wrote the book such that, although the chapters depend on each other most chapters can be read independently of each other. We sometimes even repeat material in order to allow for independent reading.

All readers should start with the chapters Introduction and Foundations. In these chapters we give an overview of LEDA and introduce the basic concepts of LEDA. We suggest that you read the chapter on foundations quickly and come back to it as needed.

The chapter on basic data types (list, stacks, queues, array, random number generators, and strings) should also be read by every reader. The basic data types are ubiquitous in the book.

Having read the chapters Introduction, Foundations and Basic Data Types, the reader may take different paths depending on interest.

Casual users of LEDA should read the chapters treating their domain of interest, and *intensive users of LEDA* should also read the chapter on implementation.

Readers interested in Data Structures should read the chapters on advanced data types, on implementation, and some of the sections of the chapter on geometric algorithms. The chapter on advanced data types treats dictionaries, search trees and hashing, priority queues, partitions, and sorted sequences, and the chapter on implementation discusses, among other things, the realization of parameterized data types. The different sections in the chapter on advanced data types can be read independently. In the chapter on geometric algorithms we recommend the section on dynamic Delaunay triangulations; some knowledge of graphs and computational geometry is required to read it.

Readers interested in Graphs and Graph Algorithms should continue with the chapter on graphs. From there one can proceed to either the chapter on graph algorithms or the chapter on embedded graphs. Within the chapter on graph algorithms the sections can be read independently.

However, the chapter on embedded graphs must be read from front to rear. Some knowledge of priority queues and partitions is required for some of the sections on graph algorithms.

Readers interested in Computational Geometry can continue with either the chapter on graphs or the chapter on geometry kernels. Both chapter are a prerequisite for the chapter on geometric algorithms. The chapter on geometry kernels requires partial knowledge of the chapter on numbers. The chapter on geometric algorithms splits into two parts that can be read independently. The first part is on convex hulls, Delaunay triangulations, and Voronoi diagrams, and the second part is on line segment intersection and polygons.

Geometric algorithms are dull without graphical input and output. The required knowledge is provided by the chapter on windows. The section on the Voronoi demo in the chapter on geometric algorithms gives a comprehensive example for the interplay between geometric data types and algorithms and the window class.

Readers interested in Algorithm Animation should read the chapter on windows and graphwin, the section on animating strongly connected components in the chapter on graph algorithms, the section on the Voronoi demo in the geometric algorithms chapter, and study the many programs in the xlman subdirectory of the demo directory.

Readers interested in Software Libraries should read the chapters on foundations, on implementation, and on documentation. They should also study some other chapters at their own choice.

Readers interested in developing a LEDA Extension Package should read the chapters on implementation and documentation in addition to the chapters related to their domain of algorithmic interest.

For all the algorithms discussed in the book, we also derive the required theory and give the proof of correctness. However, sometimes our theoretical treatment is quite compact and tailored to our specific needs. We refer the reader to the textbooks [1, 8, 16, 3, 11, 18, 14, 5, 17, 10, 13, 2, 4] for a more comprehensive view.

LEDA is implemented in C++ and we expect our readers to have some knowledge of it. We are quite conservative in our use of C++ and hence a basic knowledge of the language suffices for most parts of the book. The required concepts include classes, objects, templates, member functions, and non-member functions and are typically introduced in the first fifty pages of a C++ book [7, 9, 15]. Only the chapter on implementation requires the readers to know more advanced concepts like inheritance and virtual functions.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.

[2] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry.* Cambridge University Press, Cambridge, 1998.

[3] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press/McGraw-Hill Book Company, 1990.

[4] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer, 1997.

[5] J.H. Kingston. *Algorithms and Data Structures.* Addison-Wesley, 1990.

[6] R. Klein. *Algorithmische Geometrie.* Addison-Wesley, 1997.

[7] S.B. Lippmann and J. Lajoie. *C++ Primer.* Addison-Wesley, 1998.

[8] K. Mehlhorn. *Data Structures and Algorithms 1,2, and 3.* Springer, 1984.

[9] R.B. Murray. *C++Strategies and Tatics.* Addison-Wesley, 1993.

[10] J. Nievergelt and K.H. Hinrichs. *Algorithms and Data Structures.* Prentice Hall, 1993.

[11] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[12] T. Ottmann and P. Widmayer. *Algorithmen und Datenstrukturen.* Spektrum Akademischer Verlag, 1996.

[13] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer, 1985.

[14] R. Sedgewick. *Algorithms.* Addison-Wesley, 1991.

[15] B. Stroustrup. *The C++ Programming Language.* Addison-Wesley, 1991.

[16] R.E. Tarjan. Data structures and network algorithms. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, volume 44, 1983.

[17] C.J. van Wyk. *Data Structures and C programs.* Addison-Wesley, 1988.

[18] D. Wood. *Data Structures, Algorithms, and Performance.* Addison-Wesley, 1993.

8.1.2 LEDA Versions and Updates

Within the LEDA kernel there have been major changes in the following modules. The numbers in parentheses hint at the corresponding LEDA version:

- Most LEDA graph algorithms were improved in parallel to the evolution of the book. This lead to up to 10 times faster algorithms. (3.8)
- The data type GraphWin was extended. (3.8)
- The graph data association is much more flexible now. The information association via maps was extended to static data reservation in the graph objects (node, edge, face). (3.8)
- The multi-precision integer type was optimized. There was a major revision of the algorithmic layers as well as assembler adaptations. (3.8)
- The library is now fully prefixed. We aim for a final usage of namespaces, but use the prefixing mechanism until our compiler base supports namespaces on a large scale. (3.6)
- LEDA now offers header inclusion in full compliance with the C++-standard. (3.8)
- We now offer comparison objects which allow a more flexible handling of all order-dependent data types (dictionary, sortseq, ...) and algorithms. (3.8)

For the major improvements please refer to the manual and the corresponding sections in the LEDA book.

8.1.3 Fast Recursive Division

Investigators: Christoph Burnikel, Joachim Ziegler

Multiprecision arithmetic is the art of computing with numbers that are larger than one machine word. It is an important subject in many domains of computer science, *e.g.*, cryptography and computer algebra. Addition, subtraction, multiplication, and division of arbitrarily large numbers lie at the very heart of all computations involved in the algorithms used in these domains. For many practical applications it suffices to implement the classical school algorithms for basic arithmetic. However, if the numbers become sufficiently large, say about 200 decimal digits, it pays well to

use more sophisticated algorithms, like Karatsuba's famous method for multiplication [4], which lowers the asymptotic running time for multiplying two n-digit numbers from $\Theta(n^2)$ (by ordinary school multiplication) to $O(K(n))$ where $K(n) := n^{\log 3}$. We show in [2] how to divide a $2n$-digit number by an n-digit number in twice the time it takes to multiply two n-digit numbers with Karatsuba's method (more precisely, the running time is $2K(n) + O(n \log n)$). This compares with a running time of $\Theta(n^2)$ for school division. We also expand this method to obtain a fast algorithm for dividing arbitrary integers. An algorithm with an expected running time of $2K(n) + O(n \log n)$ was developed in [3]. This algorithm has the disadvantages that in some rare cases one has to go back to ordinary school division, *i.e.*, the worst case running time is $\Theta(n^2)$, and that it is much harder to implement.

Our algorithm performs very well in practice and yields a speedup of more than 20% with numbers in the range of 1024–2048 bits that is typical for cryptographic applications (we call this range the cryptographic range). The LEDA library [5] now ships with this algorithm integrated into its `integer` class for arbitrarily large integers. Our algorithm pays in practice for numbers with 860 bits or more. We compared our implementation with the division procedures in other public domain packages, and found that we have the fastest algorithm for integer division on a SPARC and INTEL architecture when considering all integer packages we know of. A complete description of our integer package is given in [1].

In [2] we additionally show how to achieve a running time of $3/2K(n) + O(n \log n)$ for integer division if we are not interested in the remainder of the division. We argue why one should use our method as the division method of choice in the cryptographic range. As an application of fast recursive division, we show how to speed up modular multiplication. Modular multiplication is the basic step of modular exponentiation, a method widely used in public key cryptosystems. The speedup can be achieved by combining Karatsuba multiplication and fast recursive division to perform the basic modular multiplication step. Running time experiments show that, in the cryptographic range, our algorithm is up to 40% faster than the classical modular multiplication methods.

References

- [1] C. Burnikel, S. Näher, C. Uhrig, and J. Ziegler. The design and implementation of an arbitrary precision integer library. Research report, Max-Planck-Institut für Informatik, Saarbrücken, To appear in 1999.
- [2] C. Burnikel and J. Ziegler. Fast recursive division. Research Report MPI-I-98-1-022, Max-Planck-Institut für Informatik, Saarbrücken, 1998.

[3] T. Jebelean. Practical integer division with Karatsuba complexity. *Proc. Symp. on Symbolic and Algebraic Computation, ISSAC*, pages 339–341, 1997.

[4] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics – Doklady 7*, pages 595–596, 1963.

[5] K. Mehlhorn and S. Näher. *LEDA, A Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

8.1.4 EXPCOMP — a Tool for Tuning Geometric Computation

Investigator: Stefan Funke

When computer scientists design geometric algorithms, they usually assume the availability of exact arithmetic on real numbers. Since no computer directly provides exact arithmetic on real numbers, programmers implementing these algorithms must find some substitution. Quite commonly, they

resort to floating-point arithmetic due to its support by hard- and software as well as its convenient use. The resulting programs may not behave as expected, though. There are several ways a geometric algorithm may behave when exact arithmetic is replaced by floating-point arithmetic — usually they simply crash. Hence for most problems it is crucial to ensure correctness of every predicate evaluation.

The LEDA library provides the datatypes `integer`, `rational` and `real` for exact arithmetic on integer, rational and algebraic numbers — an ideal basis for correct implementations of geometric algorithms [5, 4]. Of course, exact arithmetic with these datatypes has its cost, which is considerably higher than floating-point arithmetic. Depending on the input bit-length the arbitrary precision primitives are at least 10-100 times slower than their floating-point counterparts.

If a predicate, expressed as the sign of an arithmetic expression, is to be computed, an obvious technique to reduce this overhead is trying to decide the predicate using floating-point arithmetic first. Only if no guarantee for the correctness of the outcome can be given does one resort to arbitrary-precision arithmetic. This technique is called a *floating-point filter*.

Many predicates within LEDA are already making use of this technique — most predicates within the `rat_kernel` are optimized using a floating-point filter, and for the LEDA `real`s, filter mechanisms are incorporated into the data type itself.

Problems arise if a user wants to optimize her own predicates with floating-point filters, since writing such filter code is a non-trivial or at least cumbersome task. This situation naturally arises, if the predicate involves previously computed geometric objects and floating-point filters are to be used on all levels of computation.

Our software package EXPCOMP [2, 1, 3] (which stands for *EXPression COMPiler*) allows the programmer to easily provide floating-point filter mechanisms for her own predicates and computations without being an expert in that field. The programmer basically has to mark the appropriate code fragments using some special statements. EXPCOMP then runs as a preprocessor over the decorated source file and replaces the marked code sequences by the appropriate filtering code. Experiments with a rather involved implementation of an algorithm for computing the Voronoi Diagram of line segments and points [6] show that the use of EXPCOMP provides a considerable performance gain compared to the implementation using LEDA data types only.

EXPCOMP will be part of a forthcoming LEDA Extension Package on number types.

References

- [1] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. *International Journal of Computational Geometry and Applications — special issue*. To appear.
- [2] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. In *Proceedings of the 14th International Annual ACM Symposium on Computational Geometry (SCG-98)*, pages 175–183, Minneapolis, USA, 1998. Association of Computing Machinery (ACM), ACM Press.
- [3] S. Funke. Exact arithmetic using cascaded computation. Master's thesis, Universität des Saarlandes, 1997.
- [4] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999. The book is available at `www.mpi-sb.mpg.de/~mehlhorn`.
- [5] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. *The LEDA User manual*, 3.7 edition, 1998. See `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

[6] M. Seel. Eine Implementierung abstrakter Voronoidiagramme. Master's thesis, Universität des Saarlandes, 1994.

8.1.5 A Software Library of Dynamic Graph Algorithms

Investigator: Christos Zaroliagis

A dynamic graph algorithm is a data structure operating on a graph supporting two types of operations: updates and queries. An update is a local change of the graph and a query is a question about a certain property of the current graph. The aim of such a data structure is to use structural information about the current graph in order to handle an update faster than by the obvious solution, that is, recomputing everything from scratch with a static algorithm. Usually, queries take less time than updates, and the sequence of operations (updates and queries) is not known in advance.

Since the input of a dynamic graph algorithm is more complicated than in the static case, and static graph algorithms for basic problems like connectivity or shortest paths are very efficient as a result of decades of research, dynamic graph algorithms sometimes have to be quite sophisticated to beat the static ones in theory. Practically, the actual running times depend on many parameters including the size and type of the input graphs, the distribution of operations, and sometimes even on certain patterns in the update sequence. Consequently, in order to be able to choose the appropriate data structure for a certain application, it is usually inevitable to do some experiments with different data structures. In the best case, these experiments give some problem-specific insight, which may lead to improved algorithms and better implementations.

We have recently developed a library of dynamic data structures [1] that allows experimental comparison of different approaches with respect to inputs with specific properties. It is a joint effort of 5 groups: Univ. of Halle (D. Alberts), Max-Planck-Institute for Computer Science (C. Zaroliagis), Univ. of Rome "La Sapienza" (U. Nanni), Univ. of Rome "Tor Vergata" (G. Italiano), and Univ. of Salerno (G. Cattaneo). The library is easily adaptable and extensible. It is written in C++ and provided as a LEDA extension package (LEP). The library is available for non-commercial use from `http://www.mpi-sb.mpg.de/LEDA/friends/dyngraph.html`. It is accompanied by several demo programs, platforms on which to perform experiments, as well as correctness checkers.

The library supports several implementations of simple as well as sophisticated data structures for dynamic connectivity, dynamic minimum spanning trees, dynamic single source and all-pairs shortest paths, and dynamic transitive closure. All data structures are implemented as C++ classes derived from a common base class `dga_base`. This base class defines a common interface. There were two main problems in the design and implementation of the library.

Missing Update Operations: The algorithms usually support only a subset of all possible update operations, *e.g.*, most dynamic graph algorithms cannot handle single node deletions and insertions.

Maintaining Consistency: In an application, a dynamic graph algorithm D may run in the background while the graph changes due to a procedure P, which is not aware of D. Then there has to be a means of keeping D consistent with the current graph, because P will not use a possible interface for changing the graph provided by D, but will use the graph directly. Whether D exists or not should have no impact on P.

We decided to support all update operations for convenience. Those updates that are not supported by the theoretical background are implemented by reinitializing the data structure for the new graph. The documentation tells the users which updates are supported efficiently or not. The fact that the user calls an update that is theoretically not supported results only in a (perhaps even negligible) performance penalty. This enhances the robustness of the applications using the library or alternatively reduces the complexity of handling exceptional situations.

An obvious approach to maintain consistency between a graph and a dynamic data structure D working on that graph is to derive D from the graph class. However, this is not very flexible. In

the case when there are more than one dynamic graph data structures working on the same graph, things would get quite complicated with this approach. Instead, we use the following method motivated by the observer design pattern of Gamma *et al.* [2]. We create a new graph type `msg_graph` that sends messages to interested third parties whenever an update occurs. The base class `dga_base` of all dynamic graph algorithms is one such third party, it receives these messages and calls the appropriate update operations, which are virtual methods appropriately redefined by the specific implementations of dynamic graph algorithms.

References

- [1] D. Alberts, G. Cattaneo, G.F. Italiano, U. Nanni, and C. Zaroliagis. A software library of dynamic graph algorithms. In *Proc. Workshop on Algorithms and Experiments – ALEX'98*, pages 129–136, 1998.

  [2] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object–Oriented Software*. Addison-Wesley, 1997.

8.1.6 LEDA for Secondary Memory

Investigator: Andreas Crauser

The forthcoming LEDA extension package **Secondary memory** offers algorithmic extensions for large data sets in secondary memory. The package is called LEDA-SM and will finally become a fully integratable extension to the standard main memory code base provided by LEDA. A preliminary version is accessible at `www.mpi-sb.mpg.de/~crauser/leda-sm.html`.

8.1.7 LEDA Extension Package "Abstract Voronoi Diagrams"

Investigator: Michael Seel

The LEDA extension package **avd** implements the construction of a class of Voronoi diagrams called Abstract Voronoi Diagrams. At first it provides a framework that can be used to calculate Abstract Voronoi Diagrams in the plane. To get a program that calculates a concrete type of Voronoi diagram the user has to implement some basis operations that allow the adaptation of the framework to the concrete geometry of the problem. The framework is already adapted to the problem of the Euclidean Voronoi diagram of points and line segments in the plane.

References

- [1] M. Seel. LEP Abstract Voronoi Diagrams. `http://www.mpi-sb.mpg.de/LEDA/friends/avd.html`, 1998.

8.1.8 LEDA Extension Package "Graph Iterators"

Investigator: Marco Nissen

The LEDA extension package (LEP) **graphiterator** proposes a method for decoupling graph data structures from graph algorithms. Iterators traverse graphs in an arbitrary order. Data accessors [2] are introduced for decoupling the parameter values associated with graph objects (node or edge) from the actual algorithms. The LEP graphiterator brings ideas from the STL (Standard Template Library) and the use of LEDA together. Additionally, example algorithms (depth first search, breadth first search, strongly connected components, topological sorting, Dijkstra shortest path)

are presented. The LEP graphiterator intensively uses design patterns [1] like the iterator pattern, adapter and observer.

References

[1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object–Oriented Software.* Addison-Wesley, 1997.

[2] D. Kühl and K. Weihe. Data access templates. Konstanzer Schriften in Mathematik und Informatik Nr. 9, Universität Konstanz, Germany, May 1996. `http://www.informatik.uni-konstanz.de/Schriften`.

- [3] M. Nissen. LEP Graph Iterators. `http://www.mpi-sb.mpg.de/LEDA/friends/git.html`, 1998.

8.1.9 LEDA Extension Package "Steiner Trees"

Investigator: Ernst Althaus

The LEDA extension package **Steiner trees**, which will be finished soon, is an extension allowing calculation of exact Steiner Trees in the Plane.

Given a set of points in the plane, the Steiner Tree problem is to find a minimum-length interconnection of these points according to some geometric distance metric. The algorithms of this LEP solve the Steiner Tree problem for the rectilinear and the Euclidean metric.

The algorithms follow the branch and cut strategy for solving NP-complete problems. To formulate the problems as integer programming problems, they are transformed into the problem of finding a minimum spanning tree in a hypergraph. The transformation algorithms are due to Zachariasen [5] in the rectilinear case and to Winter and Zachariasen [4] in the Euclidean case. The algorithms for the minimum spanning tree in a hypergraph problem are due to Warme [2]. The first implementations of these algorithms were presented by Warme, Winter and Zachariasen [3]. For further information please refer to `http://www.mpi-sb.mpg.de/~althaus` and [1].

References

- [1] E. Althaus. Berechnung optimaler Steinerbäume in der Ebene. Master's thesis, Universität des Saarlandes, 1998.

[2] D. M. Warme. *Spanning Trees in Hypergraphs wiht applications to Steiner Trees.* PhD thesis, University of Virginia, 1998.

[3] Davis Warme, Pawel Winter, and Martin Zachariasen. Exact algorithms for plane steiner tree problems: a computational study. Kobenhavns universitet, Kobenhavns Universitet / Datalogisk Institut, Kopenhagen, 1998.

[4] Pawel Winter and Martin Zachariasen. Large euclidean steiner minimum trees in an hour. Kobenhavns universitet, Kobenhavns Universitet / Datalogisk Institut, Kopenhagen, 1996.

[5] Martin Zachariasen. Rectillinear full steiner tree generation. Kobenhavns universitet, Kobenhavns Universitet / Datalogisk Institut, Kopenhagen, 1996.

8.1.10 Further LEDA Extension Packages

Currently there are the following packages on our web servers which extent LEDA into special fields of algorithmic research: *avd*, *dd_geokernel*, *dynamic_graphs*, *graphiterator*, *pq_tree*, and *sd_tree*. We give a short description of recent work that has not been described above. See also `http://www.mpi-sb.mpg.de/LEDA/friends/leps.html`.

dd_geokernel implements the basic data types of higher-dimensional computational geometry: points, vectors, directions, hyperplanes, segments, rays, lines, spheres, affine transformations, and operations connecting these types. All geometric primitives are exact, *i.e.*, they do not incur rounding error (because they are implemented using rational arithmetic) and always produce the correct result. The LEP has been templatized by the arithmetic type.

pq_tree is a data structure for representing the permutations of a set U in which various subsets of U occur consecutively. Along with the data structure, efficient algorithms for manipulating PQ-trees are provided, requiring linear time in the size of the input.

sd_tree implements a data structure that provides nearest-neighbour and other kinds of search algorithms on static sets of points in two-dimensional space with Euclidean distances. The data structure of the binary search tree allows one to execute these searches in amortized constant time.

8.2 AGD: A Library of Algorithms for Graph Drawing

Investigators: Ralf Brockenauer, Carsten Gutwenger, Gunnar Klau, Petra Mutzel, René Weiskircher, Thomas Ziegler

When we started our activities in graph drawing about $4\frac{1}{2}$ years ago, our main focus was on the drawing method using planarization. Although this method produces the best drawings for many non-hierarchical practical instances (see [3]), only one software package using this method existed. (This was Giotto, developed by the research groups of Roberto Tamassia and Giuseppe di Battista. Giotto was not publically available at that time.) The reason for this lies in the fact that a great deal of theoretical and practical effort is needed to produce an implementation of the planarization method. Theoretically, many NP-hard combinatorial optimization problems arise. Moreover, one needs to understand planar graph theory concerning embeddings quite well. On the practical side, complicated algorithms, like planarity testing, combinatorial embedding, planarization routines, and planar graph drawing algorithms need to be implemented. The estimation for the amount of effort required for implementing the graph drawing method using planarization from scratch is about two to three person years.

In order to make the graph drawing method using planarization and other graph drawing methods accessible to practitioners and researchers in the field, we have developed the software library **AGD** (**A**lgorithms for **G**raph **D**rawing).

AGD contains a wide variety of graph drawing methods, in particular many new methods that we have developed in this context (see Section 7 and the last report) and tools for implementing new algorithms. The algorithms include planar graph drawing methods such as straight-line, poly-line, orthogonal, visibility, and tree drawing methods. In order to make these algorithms useful for general graphs, we provide various planarization methods ranging from heuristic to optimal algorithms. Data structures, like, *e.g.*, PQ-trees, have been especially tailored for applications in graph drawing. Users can engineer their own hybrid methods by combining the provided tools like planarization, 2-layer crossing minimization, and various shelling orders (see Figure 8.2).

Today, the planarization method is also publically available in the software library GDToolkit developed by the group of Giuseppe Di Battista (`http://www.dia.uniroma3.it/~gdt/`). The uniqueness of AGD is given by the fact that it is the only software library in the field of graph drawing that is able to solve moderate-size instances of various NP-hard combinatorial optimization

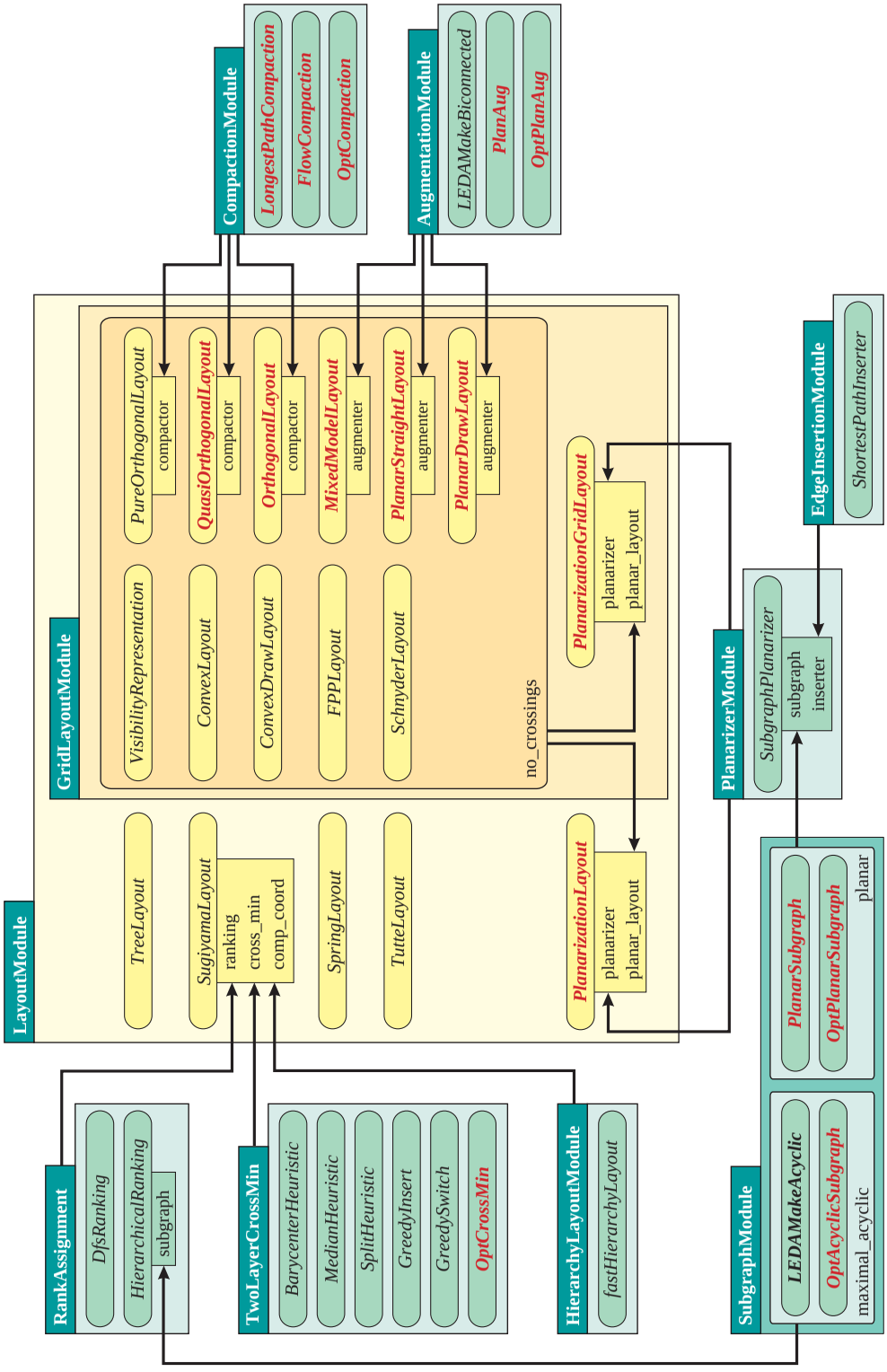


Figure 8.4: Overview of the AGD library.

problems to provable optimality within short computation time. Often, this leads to nicer drawings compared to those obtained using non-optimal methods (see Section 7).

AGD contains a new concept for the representation of algorithms that allows one to realize subtasks with exchangeable implementations. Here, the algorithms are represented as modules with a specific type, a pre- and a postcondition. Modules that can be used for a certain subtask are characterized by their type, their guaranteed precondition and their required postcondition. An AGD user can choose between the modules already contained in the library; moreover, she is free to implement and use new modules.

AGD is written in the programming language `C++` and uses the LEDA platform for combinatorial and geometric computing [7, 6]. The design of the library is based on the object-oriented features of `C++`. Graph drawing algorithms as well as combinatorial algorithms are modeled as classes. The implementations of exact optimization algorithms for NP-hard problems use the branch-and-cut system ABACUS [5]. The algorithms are implemented independent of visualization or graphics system by using a generic layout interface. A layout interface is currently available for the LEDA data type GraphWin. The open design makes AGD very easy to use and to extend.

The design of AGD is described in [2, 8] and detailed further in [1, 4]. AGD is publically available for non-commercial use via `http://www.mpi-sb.mpg.de/AGD/`. Very recently, the spin-off company Algorithmic Solutions GmbH (`http://www.algorithmic-solutions.de`) has started to provide AGD for commercial use.

AGD has been developed within the project "Design, Analysis, Implementation, and Evaluation of New Algorithms for Graph Drawing" funded by the DFG (see Section 14.2.3). Partners and group leaders of the project are: Universität Halle (Professor Dr. S. Näher), Universität zu Köln (Professor Dr. M. Jünger), Universität Passau (Professor Dr. F.-J. Brandenburg), and the Max-Planck-Institut für Informatik in Saarbrücken (Dr. P. Mutzel).

References

- [1] AGD. *AGD User Manual.* Max-Planck-Institut Saarbrücken, Universität Halle, Universität Köln, 1998. Available via `http://www.mpi-sb.mpg.de/AGD/`.
- [2] D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD-Library: A library of algorithms for graph drawing. In G. F. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE '97)*, 1997. Venice, Italy, Sept. 11-13, http://www.dsi.unive.it/~wae97/.

[3] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7:303–326, 1997.

- [4] C. Gutwenger. Design und Implementierung einer Algorithmen-Bibliothek zum Zeichnen von Graphen. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.

[5] M. Jünger and S. Thienel. Introduction to ABACUS - A Branch-And-CUt System. *Technical Report No. 97.263, Institut für Informatik, Universität zu Köln*, 1997. To appear in Operations Research Letters, 1999.

- [6] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999. The book is available at `www.mpi-sb.mpg.de/~mehlhorn`.
- [7] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. *The LEDA User manual*, 3.7 edition, 1998. See `http://www.mpi-sb.mpg.de/LEDA/leda.html`.
- [8] P. Mutzel, C. Gutwenger, R. Brockenauer, S. Fialko, G. Klau, M. Krüger, T. Ziegler, S. Näher, D. Alberts, D. Ambras, G. Koch, M. Jünger, C. Buchheim, and S. Leipert. AGD: A library of algorithms for graph drawing (poster-abstract). In S. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD '98*, volume 1547 of *Lecture Notes in Computer Science*, pages 456–457. Springer-Verlag, 1998.

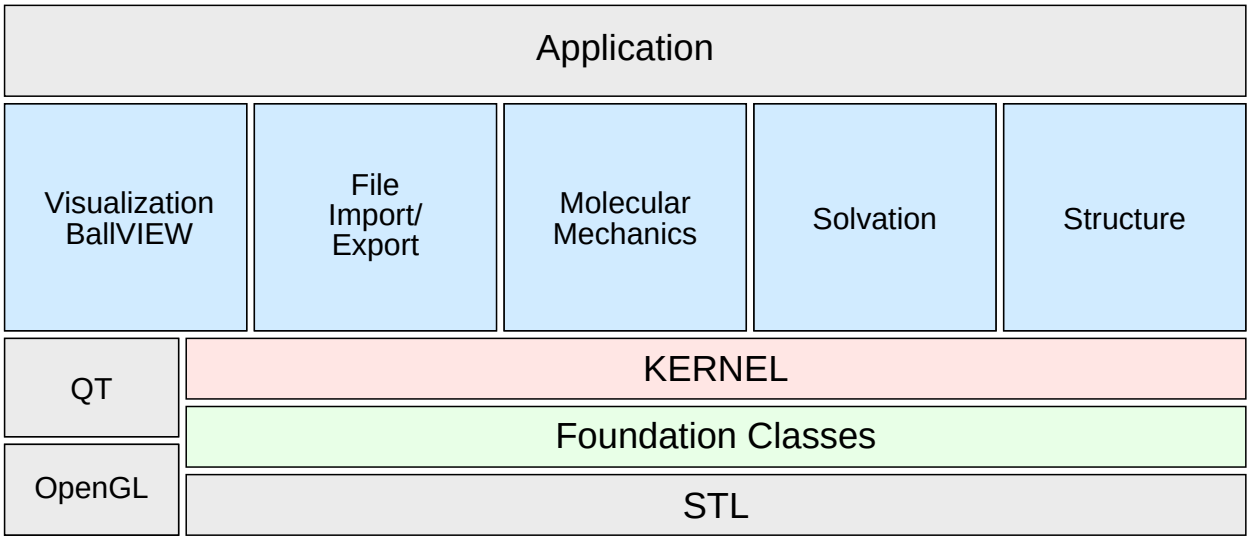


Figure 8.5: BALL components.

8.3 BALL: Biochemical Algorithms Library

The modeling and simulation of atoms, molecules, and especially biomolecules is becoming increasingly important, because it permits access to data that are not experimentally available and can save expensive biochemical experiments. The development of methods and tools in this field is hampered by a lack of state-of-the-art development tools. We performed an exhaustive search for such a tool. Commercial tools are available, but most of them are based on scripting languages (*e.g.* HyperChem [6]). There are also software development kits (SDKs) available for some commercial packages, for example for MSI's Cerius^2 [3]. However, there are no object-oriented packages available and it is impossible to create free software based on these products. Out of academia, some class libraries have developed, but they are either very specialized (*e.g.* PDBLib [4], a class library for handling Brookhaven Protein Data Bank files), or they are only remotely related to Molecular Modeling (for example SCL [7], a class library for sequences). We therefore decided to develop BALL: a Biochemical Algorithms Library.

BALL is the first object-oriented application framework in C++ that is intended for rapid software prototyping in Molecular Modeling and related areas. We decided to develop BALL to reduce the development time of our own applications (*e.g.* protein docking) and to provide the Molecular Modeling community with a state-of-the-art tool kit that might help to develop more robust, well-designed applications in less time.

BALL consists of several, inter-dependent components that are shown in Figure 8.5. The central part of BALL is the *kernel*, a set of data structures representing atoms, molecules, proteins, and so on. The kernel is implemented using the *foundation classes* that extend – and partially depend on – the classes provided by the STL (*e.g.* the vector class) and ANSI C++ (*e.g.* the string class). These three layers (STL, foundation classes, and kernel) are used by the different *basic components* of the fourth layer. Each of these basic components provides functionality for a well-defined area: *file import/export* provides support for various file formats, primarily to read and write kernel data structures. The *visualization* component BALLVIEW provides portable visualization of the kernel data structures and general geometric primitives. The *Molecular Mechanics* component contains an implementation of the AMBER95 force field [5] and support for user-defined force

fields. The *structure* component provides functionality for the comparison of three-dimensional structures, mapping these structures onto each other and searching for structural motifs. The *solvation* component primarily contains a numerical solver for the Poisson-Boltzmann equation. The behavior and properties of solvated molecules, *i.e.* molecules in solution, can be described using this equation.

A typical BALL application makes use of kernel data structures, the foundation classes, and one or more basic components; for example it uses the import/export component to read a molecule from a file, performs some simulation using the Molecular Mechanics component, and visualizes the result using BALLVIEW.

We have shown the rapid prototyping capabilities of BALL by implementing an algorithm for the three-dimensional mapping of two proteins. This algorithm was formerly implemented in the course of a Master's thesis [1]. This first implementation took about five months, whereas we could reimplement a more efficient version within a day using BALL.

The design and functionality of BALL are described in a technical report [2], a paper describing BALL has been submitted to WAE '99.

References

- [1] Jörg Becker. Allgemeine approximative Kongruenz zweier Punktmengen im R. Master's thesis, Universität des Saarlandes, 1995.
- [2] N.P. Boghossian, O. Kohlbacher, and H.-P. Lenhof. BALL: Biochemical Algorithms Library. Technical Report MPI-I-99-1-002, Max-Planck-Institut für Informatik, Saarbrücken, April 1999.

[3] Cerius^2 modeling environment. Molecular Simulations Inc., San Diego, 1997.

[4] W. Chang, I.N. Shindyalov, C. Pu, and P.E. Bourne. Design and application of PDBLib, a C++ macromolecular class library. *CABIOS*, 10(6):575–586, 1994.

[5] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A Kollman. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1995.

[6] HyperChem release 4.5. Hypercube Inc., 1995.

[7] W. Vahrson, K. Hermann, J. Kleffe, and Burghardt Wittig. Object-oriented sequence analysis: SCL – a C++ class library. *CABIOS*, 12(2):119–127, 1996.

8.4 CGAL

The area of computational geometry has developed over the last twenty years as a discipline of theoretical computer science, which drew much of its motivation from geometric applications (computer graphics, robotics, VLSI design, CAD, GIS, scientific computing). It has extracted the essential basic geometric algorithmic problems and developed efficient solutions for them. The goal of GALIA is to make the most important of these solutions and methods available to users in industry and academia in the form of a C++ library: CGAL (Computational Geometry Algorithms Library). The distinguishing features of CGAL are the careful and efficient treatment of robustness issues, the wide scope of the algorithms and data structures provided, and flexibility, extensibility, and ease of use. The development of CGAL was started in ESPRIT LTR project CGAL (project number 21957) and is continued in ESPRIT LTR project GALIA (project number 28155). The GALIA project is carried out by a consortium of seven sites: Max-Planck-Institut für Informatik, ETH Zürich (Switzerland), Freie Universität Berlin (Germany), INRIA Sophia-Antipolis (France), Martin-Luther-Universität Halle-Wittenberg (Germany), Tel-Aviv University (Israel), and Utrecht

University (The Netherlands). Utrecht University was the prime contractor for the CGAL project, Max-Planck-Institut is the prime contractor for the GALIA project.

8.4.1 Generic Programming in CGAL

Investigator: Stefan Schirra

Computational geometry has many potential application areas with different needs. As a foundation for application programs in all these areas, CGAL has to be flexible. Therefore, flexibility is a major design goal in CGAL. In [2], we discuss flexibility and the further major design goals for CGAL, which are correctness, robustness, extensibility, and ease of use, and present our approach to reach these goals. Generic programming using templates in C++ plays a central role in the architecture of CGAL. Algorithms and data structures in CGAL are generic: they work with a variety of implementations of predicates and representations of geometric objects. As illustrated in Figure 8.6, the algorithms and data structures in CGAL are parameterized by the types on which they operate. Everything that fulfills certain syntactical and semantical requirements on these types can be used with CGAL's algorithms and data structures. Of course, the geometric primitives, *i.e.* geometric predicates and basic geometric objects, provided by CGAL fulfill these requirements. The geometric primitives are usually called the *geometry kernel*. The geometry kernel of CGAL is parameterized as well. A user can choose between different coordinate representations. Currently, representations by Cartesian coordinates and representations by homogeneous coordinates are available. For both representations, the user can choose a number type, which is used to represent the coordinates and to do the arithmetic operations.

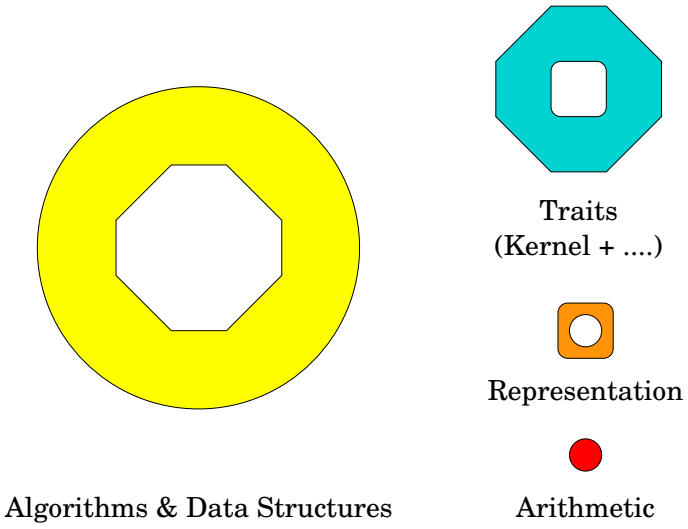


Figure 8.6: Generic programming in CGAL. CGAL algorithms and data structures (left side) are parameterized by the types on which they operate. Geometry kernels (right side) are parameterized by a coordinate representation. These representations are parameterized by a number type used to store coordinates and to do the calculations.

Generic programming using templates became popular with the Standard Template Library [3]. In [1], we argue that generic programming is especially relevant to geometric computing. In

particular, it eases exact geometric computation. For example, using an appropriate number type like the `leda_real` makes exact computation very easy. The parameterization is also a source of efficiency. In certain contexts, special primitives can be used that are less generally applicable (in terms of robust computation), but are more efficient than general-purpose solutions. The use of different number types in CGAL's geometry kernels is also discussed in Section 8.4.3.

References

- [1] H. Brönniman, L. Kettner, S. Schirra, and R. Veltkamp. Applications of the generic programming paradigm in the design of CGAL. Technical Report MPI-I-98-1-030, Max-Planck-Insitut für Informatik, 1998.

[2] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. On the design of CGAL, the computational geometry algorithms library. Max-Planck-Institut für Informatik: research report, Max-Planck-Institut für Informatik, Saarbrücken, 1998.

[3] D.R. Musser and A. Saini. *STL tutorial and Reference Guide.* Addison-Wesley, Reading, 1996.

8.4.2 Perturbations and Generic Sweep in CGAL

Investigators: Michael Seel, Mark Ziegelmann

Implementing geometric algorithms is a difficult and error-prone task. One reason for this is that most of the existing algorithms are described for non-degenerate input to simplify presentation. However, using input data from real-world applications or random input, degenerate cases (collinear points, overlapping segments, etc.) are very likely to occur. When implementing such algorithms, we are faced with the problem to identify and treat degenerate cases, which leads to additional coding and often lets the structure of the program deteriorate. If one simply does not care about degenerate cases one is often faced with incorrect output or crashes. Another approach to deal with degeneracies, which is often used in papers to state that the result also holds for general inputs, is the method of perturbation, which suggests adjusting the input by an infinitesimal amount such that degeneracies are removed. More or less general perturbation methods that have been proposed are Edelsbrunner and Mücke's Simulation of Simplicity scheme (SOS) [7], Yap's symbolic scheme [10], the efficient linear scheme of Canny and Emiris [3, 4], and the randomized scheme of Seidel [9]. For an excellent survey consult the paper of Seidel [9].

In [5] we describe a generic implementation of random linear perturbations (based on [9], which removes degeneracies with high probability) within the computational geometry software library CGAL [2]. It enables the user to perturb the input objects and hence be able to code only the original algorithm without bothering about degeneracies. Contrary to previous implementations of perturbation schemes [7, 4], this is the first general and easy-to-use implementation requiring only perturbation of the input rather than each test function.

In our experiments with planar convex hull, segment intersection, Delaunay triangulation, and 3d convex hull, we have seen that the use of our perturbation implementation introduces a medium overhead factor for the running time, which depends on the runtime fraction of the arithmetic part of an algorithm, and especially on the number type used (an overhead factor of around 50 for 10-bit `doubles` in algorithms dominated by arithmetic computation but only around 2-5 for 52-bit `integers`). The performance on highly degenerate inputs increases even more.

As a benefit we obtained simpler code since we could forget about a treatment of degeneracies (this saved about 70 of 200 lines of code in the case of the segment sweep).

We conclude that our perturbation implementation is an important tool for rapid prototyping of geometric algorithms. It enables us to implement difficult algorithms in quite reasonable time if

we do not care about a medium runtime penalty. Note however that a user of the scheme still has to plan where and how the transformation between unperturbed data and perturbed data takes place. Additionally a postprocessing step may be necessary to obtain the output corresponding to the result of a standard algorithm handling non-perturbed input including its degeneracies. See [8] for the calculation of a planar map representation of a set of intersecting segments using the perturbation scheme.

The second goal of our work was to show the applicability of a generic sweep framework [1]. This is an abstraction of the plane sweep paradigm [6] that offers a clearly structured programming concept, simplification of implementation, and checking and animation support. The framework follows the CGAL idea of generic programming via templates. Our sweep class defines a concept of a general plane sweep. If a user plugs in a model fulfilling the requirements of the concept she can use the framework to realize her sweep implementation.

The perturbed segment intersection sweep is one example of its application. Other applications are the calculation of standard and constrained triangulations.

References

- [1] U. Bartuscha, S. Näher, and M. Seel. A generic plane sweep framework. Manuscript.

[2] H. Brönnimann, S. Schirra, and R. Veltkamp (editors). The CGAL Reference Manual Release 1.1, 1998. `http://www.cs.uu.nl/CGAL/`.

[3] J. Canny and I. Emiris. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995.

[4] J. Canny, I. Emiris, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, 19(1–2):219–242, 1997.

- [5] J. Comes and M. Ziegelmann. An easy to use implementation of linear perturbations within CGAL. Technical report, Max-Planck-Institut für Informatik, 1999. To appear.

[6] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational geometry: algorithms and applications*. Springer, Berlin, 1997.

[7] H. Edelsbrunner and E. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics*, 9(1):67–104, 1990.

- [8] M. Seel and M. Ziegelmann. Sweep segments easily - a perturbed approach. Manuscript.

[9] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete and Computational Geometry*, 19(1):1–19, 1998.

[10] C.-K. Yap. Symbolic treatment of geometric degeneracies. *Journal of Symbolic Computation*, 10:349–370, 1990.

8.4.3 A Case Study on the Cost of Geometric Computing

Investigator: Stefan Schirra

When you have to implement a geometric algorithm, you will probably want to implement an efficient algorithm. But there is a bit more than choosing an efficient algorithm when you actually want to get efficient code. For example, one has to address questions like, "How do you represent the geometric objects?" or "How do you implement the predicates needed?", and one would like to know how the available options affect performance. Such implementation issues are studied in [9] for planar convex hull algorithms. The generic CGAL library, cf. Section 8.4.1, provides a unique framework for studying such issues. Since the CGAL algorithms are generic, primitives can be

easily exchanged. In our case study, we used the five planar convex hull algorithms [3, 1, 4, 5, 6, 8] available in CGAL and implemented additional algorithms [2, 7] and variations. On the side of the primitive operations, we used more than 30 different geometry kernels, among them instantiations of the Cartesian and homogeneous CGAL kernels with different number types, the rational and the floating-point geometry kernels of LEDA, a new parameterized kernel that does not use reference counting for its geometric objects, and a parameterized more "object-oriented kernel" with virtual functions for coordinate access. In total, we had about 350 convex hull algorithms to compare.

An important message for implementors of geometric algorithms might be that the cost of exact geometric computation is affordable. Another important observation is that the object-oriented kernels with virtual access functions are much slower than corresponding kernels with inlined access functions in C++. The floating-point versions of these kernels were even slower than exact kernels using floating-point filters. Furthermore, it turned out that reference counting does not pay off for the considered two-dimensional problem, where the size of the data to be copied is small. Whether or not reference counting pays off is highly affected by caching and memory effects, too.

References

[1] S. G. Akl and G. T. Toussaint. Efficient convex hull algorithms for pattern recognition applications. In *Proc. 4th IEEE Internat. Conf. Pattern Recogn.*, pages 483–487, Kyoto, Japan, 1978.

[2] K. R. Anderson. A reevaluation of an efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 7(1):53–55, 1978.

[3] A. M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Inform. Process. Lett.*, 9(5):216–219, 1979.

[4] A. Bykat. Convex hull of a finite set of points in two dimensions. *Inform. Process. Lett.*, 7:296–298, 1978.

[5] W. F. Eddy. A new convex hull algorithm for planar sets. *ACM Trans. Math. Softw.*, 3:398–403 and 411–412, 1977.

[6] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 1:132–133, 1972.

[7] C. C. Handley. Efficient planar convex hull algorithm. *Image Vision Comput.*, 3:29–35, 1985.

[8] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Inform. Process. Lett.*, 2:18–21, 1973.

- [9] S. Schirra. A case study on the cost of geometric computing. In *Proc. of ALENEX'99*, 1999. To appear.

8.4.4 Efficient Exact Geometric Computation Made Easy

Investigators: Christoph Burnikel, Rudolf Fleischer, Kurt Mehlhorn, Stefan Schirra

Geometric algorithms are usually designed for the so-called "Real RAM-model" which assumes exact real arithmetic (in the sense of mathematics). When real arithmetic is simply replaced by imprecise floating-point arithmetic, geometric algorithms that are provably correct with real arithmetic may crash or compute garbage. There are two ways to resolve this dilemma: one may either design new algorithms that work correctly even with imprecise arithmetic or implement the real RAM.

The approach to base geometric computation on the Real RAM model [3, 5, 6, 9, 10] is attractive because it allows one to use the algorithms and data structures developed under the Real RAM assumption without redesign. In [2] we show that the combination of the CGAL framework for geometric computation with the number type `leda_real` of LEDA [7, 8] provides an easy-to-use and efficient basis for exact geometric computation.

Algorithms and data structures in CGAL are parameterized by the types on which they operate. The algorithms and data structures work with *any* implementation of these types as long as they fulfill certain syntactic and semantic requirements. The algorithms are correct as long as the implementation of these types are correct. In particular, combining a CGAL-algorithm with any exact geometry kernel yields a correct program. The user may choose the geometry kernel according to her needs. The CGAL kernels are parameterized with the number type used to store coordinates and to do calculations. The use of an exact number type yields an exact kernel. The number type may be chosen according to the requirements of the application. For example, for computations inside the rational numbers, an arbitrary precision rational number type could be used with the Cartesian kernel or alternatively, an arbitrary precision integer type with the homogeneous kernel. The rational geometry kernel of LEDA also supports exact computations in the rational domain. For computations that require algebraic numbers, the number type `leda_real` can be used; it provides exact computation with addition, subtraction, multiplication, division, k-th root operations, and comparisons.

In our experiments we compare geometry kernels for geometric problems with different requirements on the number types. Whenever applicable, the rational geometry kernel of LEDA was superior to the CGAL kernels with the `leda_reals`. For problems requiring computations with algebraic numbers, however, the rational geometry kernel from LEDA cannot be used.[3] Figure 8.7 shows an example we studied, the computation of the convex hull of intersection points of circles. The experiments reported in [2] show that the use of the `leda_reals` leads to much more efficient code than the use of a comparable number type from the competing `CORE` project [4]. This is mainly due to the much better separation bound[4] [1] used by the `leda_reals`.

References

[1] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. A strong and easily computable separation bound for arithmetic expressions involving square roots. In *Proc. of the 8th ACM-SIAM Symp. on Discrete Algorithms*, pages 702–709, 1997.

- [2] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Efficient exact geometric computation made easy. In *Proceedings of 15th ACM Symposium on Computational Geometry*, 1999.

[3] S. Fortune and C. Van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Transactions on Graphics*, 15(3):223–248, 1996.

[4] V. Karamcheti, C. Li, I. Pechtanski, and C. Yap. A core library for robust numeric and geometric computation. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, 1999.

[5] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. *ACM Transactions on Graphics*, 10(1):71–91, 1991.

[6] K. Mehlhorn and S. Näher. The implementation of geometric algorithms. In *13th World Computer Congress IFIP94*, volume 1, pages 223–231. Elsevier Science B.V. North-Holland, Amsterdam, 1994.

[3] A major reorganization of the code would be required to replace the algebraic sign computations by a combination of rational computations.

[4] A separation bound for an arithmetic expression is a positive lower bound on the absolute value of the expression, if the value of the expression is non-zero.

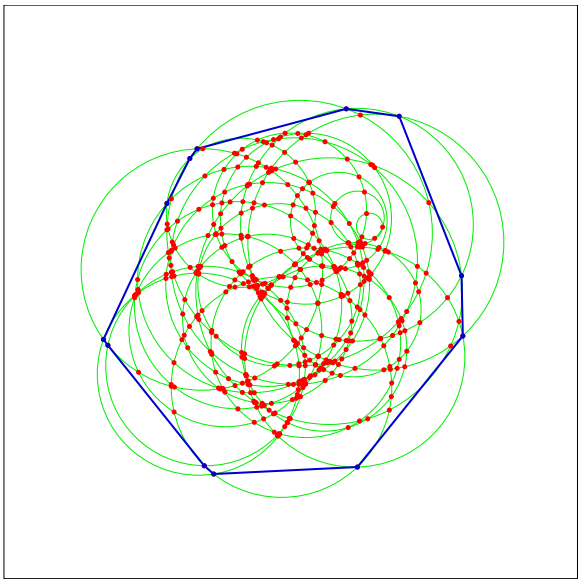


Figure 8.7: A geometric problem involving non-rational computations: convex hull of intersection points of circles.

- [7] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999. The book is available at `www.mpi-sb.mpg.de/~mehlhorn`.
- [8] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. *The LEDA User manual*, 3.7 edition, 1998. See `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

[9] C. Yap. Towards exact geometric computation. *Comput. Geom. Theory Appl.*, 7:3–23, 1997.

[10] C. K. Yap and T. Dubé. The exact computation paradigm. In D.Z. Du and F. Hwang, editors, *Computing in Euclidean Geometry*, pages 452–492. World Scientific Press, 1995. 2nd edition.

9 Visitors

Since March 1997, a total of 57 researchers have visited our group.

Prof. Foto Afrati National Technical University Athens, Greece	04/06/97	
Prof. Dr. Helmut Alt FU Berlin, Germany	03/12/97	
Nancy Amato Texas A& M University, USA	17/07/97 -	30/07/97
Arne Anderssen Lund University, Sweden	09/06/97 -	17/06/97
Dr. Lars Arge Duke University, Durham, USA	17/05/99 -	21/05/99
Boris Aronov Polytechnic University Brooklyn, New York, USA	18/06/97 -	22/08/97
Prof. Sanjiv Arora Princeton University Princeton, USA	02/08/97 -	29/08/97
Therese Biedl Rutgers University Piscataway, USA	08/01/97 -	10/01/97
Hervé Brönnimann INRIA Sophia-Antipolis, France	20/08/97 -	22/08/97
Dr. Roger Butenuth Universität Paderborn	11/02/98	
Dr. Bogdan Chlebus Uniwersytet Warszawski Warszawa, Poland	19/05/97 -	21/05/97
Prof. Maxime Crochemore Université de Marne-la-Vallée Noisy-le-Grand, France	21/05/97 -	24/05/97
Artur Czumaj	10/03/97 -	21/03/97
Universiät-GH Paderborn, Germany	21/06/97 -	24/06/97
Prof. Sajal Das University of North Texas Denton, USA	04/03/98	
Dr. Tamal Dey	06/07/98 -	30/11/98
Walter Didimo Università degli Studi di Roma La Sapienza, Italy	14/10/98 -	31/03/99
Prof. Yeffim Dinitz Technion, Haifa, Israel	28/10/97 -	26/12/97
Prof. Peter Eades University of Newcastle, Australia	01/02/99 -	13/02/99

Prof. Dr. Hubert de Fraysseix	20/07/98 -	26/07/98
EHESS, Paris, France		
Dr. Daniele Frigioni	15/04/97 -	31/03/98
Universitá di Roma "La Sapienza"	20/07/98 -	27/07/98
Roma, Italy		
Pierre-Marie Gandoin	01/09/98 -	30/09/98
INRIA Sophia Antipolis, France		
Prof. Dr. Naveen Garg	27/05/98 -	23/07/98
Indian Institute of Technology		
New Delhi, India		
Dr. Leszek Gasieniec	25/07/98 -	01/08/98
University of Liverpool		
Liverpool, UK		
Joachim Giesen	03/05/99 -	04/05/99
ETH Zürich, Switzerland		
Prof. M.X. Goemans	04/12/97 -	01/12/97
MIT, Cambridge, USA		
Prof. Michael Goodrich	17/05/99 -	21/05/99
The John Hopkins University		
Baltimore, USA		
Dr. Sumantha Guha	15/10/98 -	30/10/98
Calcutta, India		
Magnus Halldorsson	23/08/97 -	29/08/97
University of Iceland		
Reykjavik, Iceland		
Prof. Dan Halperin	18/08/97 -	23/08/97
Tel Aviv University		
Tel Aviv, Israel		
Dr. Susan Hert	15/06/98 -	17/08/98
Knox College, Galesburg, USA		
Seok-Hee Hong	01/02/99 -	13/02/99
Women's University of Seoul		
Korea		
Prof. Giuseppe Italiano	15/07/97 -	15/08/97
Universitá di Venezia, Italy		
Ben Juurlink	21/09/97 -	26/09/97
Leiden University, The Netherlands		
Prof. Howard Karloff	18/08/97	
Georgia Institute of Technology, USA		
Prof. John Kececioglu	16/06/97 -	11/07/97
University of Georgia		
Athens, USA		
Prof. Alexander Kelmans	23/08/98 -	30/08/98
Rutgers University		
New Brunswick, USA		
Prof. Dr. Leonid Khachiyan	05/08/98 -	12/08/98
Rutgers University		
New Brunswick, USA		

Samir Khuller University of Maryland College Park, USA	14/08/97 -	16/08/97
Prof. Ludek Kucera Charles University, Praha	06/12/98 -	11/12/98
Dr. Stefano Leonardi	12/05/97 -	31/10/97
Universitá di Roma "La Sapienza" Italy	09/02/98 -	30/06/98
Prof. Anil Maheshwari Tata Insitute of Fund. Res. Bombay, India	03/08/98 -	23/08/98
Robert M. Münch SCRAP EDV-Anlagen GmbH Karlsruhe, Germany	15/01/98 -	16/01/98
Prof. Stefan Näher Universität Halle, Germany	15/03/99 -	30/04/99
Prof. Manfred Padberg New York University, USA	30/11/98 -	03/12/98
Sylvain Pion INRIA Sophia Antipolis, France	20/08/97 -	22/08/97
Dr. Venkatesh Raman The Institute of Mathematical Sciences Chennai, India	27/02/98 -	27/03/98
Dr. Desh Ranjan New Mexico State University Las Cruces, USA	10/07/98 -	15/08/98
Prof. Ed Reingold University of Illinois at Urbana-Champaign USA	01/12/99	
Dr. Bernhard Schölkopf GMD-FIRST, Berlin, Germany	19/03/99 -	20/03/99
Prof. Eljas Soisalon-Soininen Helsinki University of Technology Finland	01/02/99 -	30/06/99
Boris Steipe München, Germany	21/01/98 -	22/01/98
Jens Stoye DKFZ Heidelberg, Germany	23/11/98	
Dr. Mario Szegedy AT&T Labs. Research Florham Park, USA	21/08/98	
Tuomo Takkula Chalmers University Goeteborg, Sweden	15/07/98 -	16/07/98
Prof. Roberto Tamassia Brown University Providence, USA	14/07/97 -	19/07/97

Prof. Mikkel Thorup University of Copenhagen, Denmark	30/06/97 -	01/08/97
Prof. Vijay Vazirani Georgia Institute of Technology Atlanta, USA	29/08/97 -	31/08/97
Dr. Christos Zaroliagis King's College, University of London United Kingdom	22/02/99 -	26/02/99

10 Journal and Conference Activities

10.1 Editorial Positions

Kurt Mehlhorn has been an editor of *Algorithmica* (since 1985), *Computational Geometry: Theory and Applications* (since 1990), *Information and Computation* (since 1985), *International Journal of Computational Geometry & Applications* (since 1990), *Inernational Journal of Discrete and Computational Geometry* (since 1988), and *SIAM Journal on Computing* (since 1988).

Petra Mutzel was guest editor for the *Journal of Graph Algorithms and Applications (JGAA)* for the Special Issue on Graph Drawing '97 (joint with Professor Dr. G. Di Battista)

10.2 Conference and Workshop Positions

10.2.1 Membership in program committees

Susanne Albers: 40th Annual Symposium on Foundations of Computer Science (FOCS), New York City, USA, 1999.
16th International Symposium on Theoretical Aspects of Computer Science (STACS), Trier, Germany, 1999.
25th International Colloquium on Automata, Languages and Programming (ICALP), Ålborg, Denmark, 1998.
3rd Workshop on Randomized Parallel Computing (WRPC), Orlando, USA, 1998.
Workshop on On-Line Algorithms, Udine, Italy, 1998.

Kurt Mehlhorn: 2nd Workshop on Algorithm Engineering (WAE), Saarbrücken, Germany, 1998.
6th Annual European Symposium on Algorithms (ESA), Venice, Italy, 1998.
5th Annual International Computing and Combinatorics Conference (COCOON), Tokyo, Japan, 1999.
2nd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Berkeley, USA, 1999.

Petra Mutzel: 7th International Symposium on Graph Drawing (GD '99), Prag, 1999

10.2.2 Membership in organizing committees

Klaus Jansen: First Workshop on *Approximation Algorithms for Combinatorial Optimization Problems* APPROX'98, University of Ålborg, Denmark, July 1998.

Kurt Mehlhorn: 2nd Workshop on Algorithm Engineering (WAE), Saarbrücken, Germany, 1998.

Petra Mutzel: Fifth Annual Graph Drawing Contest, held in conjunction with the 1998 Graph Drawing Symposium in Montreal, Canada, 1998.
Sixth Annual Graph Drawing Contest, held in conjunction with the 1999 Graph Drawing Symposium in Prag, 1999.

Marina Papatriantafilou, Philippas Tsigas: School on "Distributed Computing", Max-Planck-Institute für Informatik, Saarbrücken, September 1997. The school was supported by the ALCOM-IT project.

Jop Sibeyn: 7th Workshop on Algorithms for Future Technologies (ALTEC–VII), Max-Planck-Institute für Informatik, Saarbrücken, on May 1997. More information can be found at `http://www.mpi-sb.mpg.de/~jopsi/altec.html`.

Christos Zaroliagis: 2nd Workshop on Algorithm Engineering (WAE'98), Saarbrücken, Germany. More information about the workshop can be found at `http://www.mpi-sb.mpg.de/~wae98`.

11 Teaching Activities

The group contributes intensively to the curriculum of the Department of Computer Science at the Universität des Saarlandes. We teach core courses (like "Praxis des Programmierens", "Datenstrukturen und Algorithmen") and specialized courses. The details follow.

Winter Semester 1997/98

LECTURES:
Datenstrukturen und Algorithmen (K. Mehlhorn, R. Fleischer)
Scheduling (G. Schmidt, S. Albers)
Computational Molecular Biology (H.-P. Lenhof, O. Kohlbacher, P. Müller, K. Reinert)
Advanced C++ (S. Schirra)
SEMINARS:
Ganzzahlige Optimierung (P. Mutzel, G. Klau, R. Weiskircher)
PROJECT CLASSES:
Implementierung von dynamischen Graphenalgorithmen (C. Zaroliagis)
Visualisierung von Online-Algorithmen (R. Fleischer)
Implementierung paralleler Algorithmen mit Hilfe der PAD Bibliothek (J. Träff)

Summer Semester 1998

LECTURES:
Paralleles Programmieren (P. Sanders)
Advanced C++ (S. Schirra)
SEMINARS:
Randomisierte Algorithmen (J. Sibeyn)
PROJECT CLASSES:
Visualisierung von Online-Algorithmen (R. Fleischer)
Implementierung geometrischer Algorithmen (S. Schirra)
Erläuterung paralleler Algorithmen mittels Applets (J. Sibeyn)

Winter Semester 1998/1999

LECTURES:
Praxis des Programmierens (P. Sanders)
Komplexitätstheorie (K. Mehlhorn, L. Porkolab, S. Funke, M. Ziegelmann)
Algorithmen und Datenstrukturen (R. Fleischer, J. Sibeyn)
SEMINARS:
Bioinformatik (H.-P. Lenhof, S. Burkhardt)
PROJECT CLASSES:
Implementierung geometrischer Algorithmen (S. Schirra)
Algorithmen für große Datenmengen (K. Mehlhorn, A. Crauser, M. Ziegelmann)
Algorithmen zum Zeichnen von Graphen (P. Mutzel)

Summer Semester 1999

LECTURES:
Optimierung (S. Albers, R. Fleischer)
Parallele Algorithmen (P. Sanders, J. Sibeyn)
PROJECT CLASSES:
Bioinformatik (H.-P. Lenhof, S. Burkhardt)
SEMINARS:
Angewandte Algorithmische Geometrie (S. Schirra, E. Ramos)

Our group offers continually an advance course, called "Selected Topics in Algorithms", on various (advanced) topics in algorithms and complexity. This course is actually a sequence of mini-courses on graduate-level; each mini-course is taught by a set (usually singleton) of instructors which are group members and/or visitors. The advance course is mainly intented to our PhD students, but it is also attended by many members of the group. The course is organized by Stefan Schirra. Topics treated from April 1997 until March 1999 were:

Functional Data Structures (G. Brodal)
Concentration of Measure for Computer Science Applications (D. Dubhashi)
Parameterized Complexity (V. Raman)
Exact and Approximate Nearest Neighbor Queries in Euclidean Space (E. Ramos)
Random Number Generators (C. Rüb)
Parallel Heuristic Search: Algorithms, Analysis and Applications (P. Sanders)
Elementary Constructions of Expander Graphs Using Algebraic Graph Theory (O. Scheja)
Finding Paths and Cycles in Graphs (C.R. Subramanian)

Diploma Theses

During the last two years, the following 29 diploma theses have been completed under guidance of members of our group.

Werner Backes: Berechnung kürzester Gittervektoren, 1998.
Ralf Brockenauer: Separierung von Kuratowski-Ungleichungen für das größte planare Untergraphenproblem, 1997.
Thomas Feld: Analyse und Implementierung von Algorithmen zum Minimum Cost Circulation Problem, 1998.
Sergej Fialko: Das planare Augmentierungsproblem, 1997.
Christoph Gast: Das Maximum-Weight-Trace-Problem bei multiplem Sequenz-Alignment, 1997.
Frank Guillaume: Paralleles List Ranking, 1997.
Holger Kappel: Eine Methode zur Berechnung von Vorzeichen ganzzahliger Determinanten, 1998.
Björn Kettner: Eine Implementation von k-Server-Algorithmen, 1998.
Jochen Könemann: Fast combinatorial algorithms for packing and covering problems, 1997.
Klaus Kursawe: Exploration von geometrischen Umgebungen mit Hindernissen, 1998.
Carsten Kwappik: Exact Linear Programming, 1998.
Thorsten Lauer: Design und Implementierung eines Testmanagers für LiDIA, 1998.
Stefan Leinenbach: Eine effiziente Implementierung des Datentyps Polyeder, 1997.
Erwin Margewitsch: Parallele Berechnung elektrostatischer Wechselwirkungen für synthetische Polymere, 1997.
Tobias Miller: Implementation and Experimental Evaluation of Dynamic Transitive Closure Algorithms, 1998.

Thomas Mueck: Implementation of Hammock Decomposition with Application to Shortest Path Problems, 1998.
Matthias Müller: Ein Simulator für Prozessornetzwerke, 1998.
Martin Nest: Vergleich von praxisnahen seriellen und parallelisierten Verfahren zur Bestimmung der Schnittpunkte von Liniensegmenten in der Ebene, 1998.
Marco Nissen: Graph Iterators: Decoupling Graph Structures from Algorithms, 1998.
Fred Oberhauser: Arithmetik der Transduktoren, 1997.
Martin Reinstädtler: Verlustfreie Datenkompression mit selbstorganisierenden Listen, 1998.
Hein Röhrig: Tree Decomposition: A Feasability Study, 1998.
Bianca Schröder: Upper and Lower Bounds for Basic Scheduling Problems, 1998.
Tillmann Seidel: Paralleles List Ranking, 1997.
Henrik Stormer: Ein Programm zum visuellen Erlernen von Graphalgorithmen, 1998.
René Weiskircher: 2-Schicht-Planarisierung bipartiter Graphen, 1997.
Michael Wissen: Automatisiertes Zeichnen von Zustandsdiagrammen, 1998.
Kurt Ziegenbein: Bewertung verschiedener paarweiser Alignment-Methoden, Ersetzungsmatrizen und Gap-Funktionen, 1997.

12 Dissertations, Habilitations, and Offers for Associate Professorships

12.1 Dissertations

Completed:
Finkler, U.: Design of Efficient and Correct Algorithms: Theoretical Results and Runtime Prediction of Implementations in Practice, August 1997.

In preparation:
Bast, H.: Provably optimal scheduling of irregular parallel loops.
Crauser, A.: External Memory Algorithms in Theory and Practice.
Gergov, J.: Approximation Algorithms for Dynamic Storage Allocation.
Meyer, U.: Parallel algorithms for large data sets.
Müller, P.: Parallel Molecular Dynamics Simulations for Synthetic Polymers.
Priebe, V.: Probabilistic analysis of combinatorial algorithms.
Reinert, K.: A polyhedral approach to sequence alignment problems.
Schilz, T.: Effiziente Algorithmen für das verteilte Rechnen auf Workstationclustern.
Seel, M.: Intersection of Polyhedra in 3-Space.
Ziegler, T.: Crossing Minimization in Automatic Graph Drawing.

12.2 Habilitations

Susanne Albers
Rudolf Fleischer
Petra Mutzel
Jop Sibeyn

The habilitation procedures of Hans-Peter Lenhof and Stefan Schirra are on-going.

12.3 Offers for Associate Professorships

Susanne Albers:
Universität Paderborn, 1999.
Universität Trier, 1999.

Rudolf Fleischer:
University of Waterloo, Canada, 1999.

Torben Hagerup:
Universität Trier, 1997.
Universität Frankfurt, 1997.

Klaus Jansen:
Universität Kiel, 1999.

Petra Mutzel:
Technische Universität Wien, Austria, 1999.

Christos Zaroliagis:
University of Patras, Greece, 1999.

13 Organization of our Group

The group meets two to four times a week at 1.30 pm.

On Monday and Wednesday (1.30 - 2.15) we have our noon seminar. It lasts about 45 minutes and is reserved for presentations of new results and ongoing research. We also ask our guests to give presentations in the noon seminar.

On Tuesday and Thursday (1.30 - 3.00) we run the "Selected Topics in Algorithms" advance course. This course is reserved for two to four week intensive treatments of subjects of current interest and it is organized by Stefan Schirra. More information can be found in Section 11.

There are also two other meetings:

(a) The "group-meeting" that runs on a monthly basis and in which all the members of the group participate to discuss various topics regarding the group and to be informed about several other activities.

(b) The "research-associates' meeting" that runs on a 2 to 3 weeks basis and makes the basic decisions concerning the scientific strategy of the group.

For several directions that are studied in the institute, there are "Special Interest Groups". Such groups may have different natures, but typically meet every few weeks, to present some work within the scope of interest and to discuss this in some detail.

Presently groups in the following areas are active:
Approximation and Online Algorithms (contact Rudolf Fleischer)
Automatic Graph Drawing (contact Petra Mutzel)
Optimization (contact Zeev Nutov)
Computational Geometry (contact Edgar Ramos and Stefan Schirra)
Software Engineering (contact Peter Sanders)
Aspects of External Computing (contact Jop Sibeyn).

14 Cooperations

We start with local cooperations. There is cooperation within the institute and with colleagues in the computer science department. The cooperation with Raimund Seidel has led to joint publications. The cooperation with AG2 in the area of combinatorial optimization has led to a PhD thesis for which Kurt Mehlhorn was the co-advisor. We expect considerable cooperation with Hans-Peter Seidel's group and also see a potential for cooperation with Henzinger's group.

Our Computational Molecular Biology group cooperates in several projects with the Theoretische Bioinformatik Gruppe of the Deutsches Krebsforschungszentrum (DKFZ) in Heidelberg (Dr. Martin Vingron), with the Forschungsstelle der Max-Planck-Gesellschaft für die Enzymologie der Proteinfaltung in Halle (Dr. Peter Bayer), with the Max-Planck-Institut für Molekulare Physiologie in Dortmung (Dr. Axel Scheidig), with the Institut für Biopharmazie und Pharmazeutische Technologie der Universität des Saarlandes (Prof. Dr. Claus-Michael Lehr), with the Institut für Neue Materialien in Saarbrücken (Prof. Dr. Helmut Schmidt), with the Institut für Humangenetik der Universität des Saarlandes (Prof. Dr. Eckhart Meese), and with the company Across Barriers GmbH in Saarbrücken (Dr. Ellen Haltner).

We work together with NEC C&C Research Laboratories, Sankt Augustin, on collective communication routines for parallel processing, and with Philips Reserach Eindhoven on using parallel disks in multimedia servers.

At an institutional level we are involved in several research projects with various sources of funds. The projects are funded by the European Union, the German government through DFG and BMBF, and by Industry. The details are given in the rest of this section.

14.1 Projects funded by the European Union

14.1.1 ALCOM-IT

ALCOM-IT (ALgorithms and COMplexity in Information Technology) is an ESPRIT IV Long Term Research project involving 12 partners from 9 different countries. ALCOM-IT and CGAL are off-shoots of the ALCOM I and II projects. ALCOM-IT differs form its predecessors in that there is a stronger focus and greater emphasis on the applied part of research. The aim is to bridge the gap between research in the field of algorithms and applications in information technology, and to play for European industry the role that the strong algorithm groups within IBM Research and Bell Labs play for their companies.

The project was originally planned to run from January 1, 1996 to December 31, 1998, but was recently extended to June 30, 1999, and is coordinated by Prof. Giorgio Ausiello. Rudolf Fleischer is our local contact person.

The ALCOM-IT project brings together researchers from major European institutions. The partners and group leaders are:

University of Århus, Århus, Denmark (Prof. E.M. Schmidt)
Universitat Politecnica Catalunya, Barcelona, Spain (Prof. J. Diaz)
Universität zu Köln, Köln, Germany (Prof. M. Jünger)
EHESS, Paris, France (Directeur P. Rosenstiehl)
INRIA-Paris, Rocquencourt, France (Dr. P. Flajolet)
Universität-GH Paderborn, Paderborn, Germany (Prof. B. Monien)
Computer Technology Institute, Patras, Greece (Prof. P. Spirakis)
Universitá di Roma "La Sapienza", Roma, Italy (Prof. G. Ausiello)

Utrecht University, Utrecht, The Netherlands (Prof. J. van Leeuwen)
University of Warwick, Coventry, United Kingdom (Prof. M. Paterson)
ETH Zürich, Zürich, Switzerland (Prof. G. Gonnet)
Max-Planck-Institut für Informatik, Saarbrücken, Germany (Prof. K. Mehlhorn)

On January 15–17, 1998, the annual review workshop of ALCOM-IT was held at the Max-Planck-Institut für Informatik in Saarbrücken.

14.1.2 ALTEC

The ALTEC (ALgorithms for future TECnologies) project was financed by the EU, and had the purpose of joining the expertise of a number of Western and Central/Eastern European research groups while establishing a network of research and cooperation. It aimed at the further development of algorithm-based techniques for future information processing technologies.

The first period of the project (Cooperative Action IC 1000) had expired on March 1996. In the fall of 1996, we entered a second period. This was a so-called "Keep In Touch" project. It built further on the achievements of ALTEC: its main objective was to support the existence of the ALTEC co-operative network. On the technical level, the goal was to stimulate joint research and cooperation in the areas of parallel architectures, parallel algorithms and their implementation, and the underlying communication networks of parallel computers. More generally, it stimulated research on the effective exploitation of novel programming techniques and programming support environments in the area of high performance computing and networking (HPCN). The project ended in 1998.

Jop Sibeyn was our local contact person. The ALTEC web-page is maintained by our institute, and provides more information about the project. It can be found at `http://www.mpi-sb.mpg.de/~jopsi/altec.html`.

14.1.3 CGAL and GALIA

The goal of the ESPRIT Long Term Research projects CGAL (Constructing a Geometric Algorithms Library) and GALIA (Geometric ALgorithms for Industrial Applications) is to make the most important of solutions and methods developed in the field of computational geometry available to users in industry and academia in a software library. This software library is called CGAL (Computational Geometry Algorithms Library). The work on CGAL has been started in the CGAL-project (October 1996 til June 1998) and is now continued in the GALIA project (November 1995 til May 2000). The projects have their roots in the ALCOM projects. While Utrecht University was prime contractor for the CGAL project, Max-Planck-Institut für Informatik is prime contractor for GALIA.

Partners and group leaders of the CGAL project are:

Utrecht University, Utrecht, The Netherlands (Prof. M. Overmars, Dr. M. de Berg, Dr. R. Veltkamp)
ETH, Zürich, Switzerland, (Prof. E. Welzl, Prof. P. Widmayer, Prof. J. Nievergelt)
Freie Universität Berlin, Germany (Prof. H. Alt)
INRIA Sophia-Antipolis, Sophia-Antipolis, France (Dr. J.-D. Boissonnat, Dr. A. Fabri)
RISC, Linz, Austria (Dr. S. Stifter)
Tel Aviv University, Israel (Prof. D. Halperin, Prof. M. Sharir)
Max-Planck-Institut für Informatik, Saarbrücken, Germany (Prof. K. Mehlhorn, Dr. S. Schirra)

Partners and group leaders of the GALIA project are:

Max-Planck-Institut für Informatik, Saarbrücken, Germany (Prof. K. Mehlhorn, Dr. S. Schirra)
ETH, Zürich, Switzerland, (Prof. E. Welzl, Prof. P. Widmayer)
Freie Universität Berlin, Germany (Prof. H. Alt)
INRIA Sophia-Antipolis, Sophia-Antipolis, France (Dr. J.-D. Boissonnat, Dr. M. Yvinec)
Martin-Luther-Universität Halle, Germany (Prof. S. Näher)
Tel Aviv University, Israel (Prof. D. Halperin)
Utrecht University, Utrecht, The Netherlands (Prof. M. Overmars, Dr. R. Veltkamp)

14.1.4 TADEQ

TADEQ is a research and development project in the EU Esprit Programme, R&D in the domain of Software Technologies concerning the topic of Statistical Systems. TADEQ is a short form for "A Tool for the Analysis and Documentation of Electronic Questionnaires". National Statistical Institutes, research institutes, and commercial marketing research organizations are making an increased use of computer-assisted interview (CAI) systems for collecting survey data. The growing possibilities of computer hardware and software have made it possible to develop very large, and complex electronic questionnaires. It has become more and more difficult for developers, interviewers, supervisors, and managers to keep control of the content and structure of CAI instruments.

The TADEQ project proposes to develop a tool to make a human-readable presentation (on paper or electronically in hypertext format) of the electronic questionnaire. Such a tool should not only provide a useful documentation of the contents and structure, but also help to analyze the questionnaire, and report possible sources of problems in its structure.

Partners and group leaders of the project are:

Statistics Netherlands (Prof. Dr. J. Bethlehem)
Max-Planck-Institut für Informatik (Dr. P. Mutzel)
Office of National Statistics, United Kingdom (T. Manners)
Statistics Finland (V. Kuusela)
Instituto Nacional de Estatistica, Portugal (J.C. Marques Nunes)

14.2 Projects funded by DFG

Four projects are funded by the German National Science Foundation (DFG – Deutsche Forschungsgemeinschaft). The first project was supported by the special research area program (SFB – Sonderforschungsbereich) of DFG, while the other three are supported by a special program of DFG (DFG-Schwerpunktprogramm).

14.2.1 SFB 124 VLSI–Entwurfsmethoden und Parallelität

The SFB 124 was a special research effort on VLSI design methods and parallelism. The part of the project in which our group has been involved has been centered around more practical aspects of parallel systems: design, programming (language and implementation aspects) and applications. The SFB 124 was initiated in 1983, and has ended by December 1997.

For the last, 3 year period of the project the partners were as follows:

Universität Kaiserslautern (Prof. Zimmermann, Dr. Schuermann)
Universität des Saarlandes (Professors: Buchmann, Hotz, Loeckx, Paul, Wilhelm; Dr. Ruenger)
Max-Planck-Institut für Informatik (Prof. Mehlhorn)

14.2.2 Molecular Dynamics Simulations of Synthetic Polymers

The project was part of the DFG-Schwerpunktprogramm "Efficient Algorithms for Discrete Problems and their Applications". The goal of the project was the development and implementation of efficient parallel algorithms for MD-simulations of synthetic polymers. It started in 1995 and ended at the end of 1997.

Partners and group leaders of the project were:

Max-Planck-Institut für Polymerforschung, Mainz, (Dr. B. Jung)
Max-Planck-Institut für Informatik, Saarbrücken, (Dr. C. Rüb, Dr. H.-P. Lenhof)

14.2.3 Graph Drawing

We are involved in the cluster "Efficient Algorithms for Discrete Problems and Their Applications" with the project "Design, Analysis, Implementation, and Evaluation of New Algorithms for Graph Drawing". The aim of this project is mainly to develop new techniques for drawing graphs. Moreover, we have designed a software library (AGD-LIB), which is independent of the used graph editor (*e.g.*, GraphWin or Graphlet) and contains modules of graph drawing algorithms. For more information, see Section 8.2.

The project started in 1995, and will be running until the end of 1999. Hopefully, we can get a prolongation for one more year this fall.

Partners and group leaders of the project are:

Universität Halle (Prof. Dr. S. Näher)
Universität Köln (Prof. Dr. M. Jünger)
Universität Passau (Prof. Dr. F. Brandenburg)
Max-Planck-Institut für Informatik (Dr. P. Mutzel)

14.2.4 Protein-Protein-Docking

The project is part of the DFG-Schwerpunktprogramm "Computer science methods for the analysis and interpretation of large genomic data sets". The goal of the project is the development and implementation of algorithms for the protein-protein-docking problem. It started in October 1998 and funds one research assistant position for two years.

Partners and group leaders of the project are:

Max-Planck-Institut für Molekulare Physiologie, Dortmund (Dr. A. Scheidig)
Forschungsstelle der Max-Planck-Gesellschaft für die Enzymologie der Proteinfaltung, Halle (Dr. P. Bayer)
Max-Planck-Institut für Informatik, Saarbrücken (Dr. H.-P. Lenhof)

14.3 BMBF Grant

The German Ministry of Education, Science, and Technology founded the mathematical program "Mathematical Methods for Problem Solving in Trade and Industry". Promoted are projects that are developing new mathematical methods for real practical problems in direct cooperations with partners from trade and industry. We got a BMBF grant, that is a position for an associate researcher for the duration of three years starting in July 1997. In our project we are investigating the (map) labeling problem in connection with the drawing problem, since for the drawing of the finite state machines for SIEMENS, the relatively big labels of the states and the state transitions

lead to serious problems. Our industrial partner is SIEMENS AG. The project leader is Petra Mutzel.

14.4 Cooperations with Industry

We are cooperating with SIEMENS on a project on automatic drawing of finite state machines that describe the control of computer integrated manufacturing processes. Here, we show that our graph drawing techniques based on planarization methods are indeed practical. The SIEMENS AG is sponsoring a PhD fellowship to us for three years. The project started in 1996, and the project leader is Petra Mutzel. Our contact person at SIEMENS is Dr. U. Lauther.

For the LEDA project the main industrial partner is Algorithmic Solutions GmbH. This partnership is "legalized" by a contract between the Max-Planck-Gesellschaft and Algorithm Solutions GmbH which gives Algorithm Solutions GmbH the right to market LEDA and regulates the royalties that have to be paid by Algorithm Solutions GmbH in return.

15 Recent Publications

Books

[1] J. Keller, C. W. Keßler, and J. L. Träff. *Practical PRAM Programming.* Wiley International, 1999. Book in preparation, scheduled for late 1999.

[2] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999. The book is available at `www.mpi-sb.mpg.de/\~mehlhorn`.

In Journals and Book Chapters

[1] S. Albers and H. Koga. New on-line algorithms for the page replication problem. *Journal of Algorithms*, 27:75–96, 1998.

[2] S. Albers and S. Leonardi. Online algorithms. *ACM Computing Surveys, Electronic Issue*, 1999.

[3] S. Albers and M. Mitzenmacher. Revisiting the counter algorithms for list update. *Information Processing Letters*, 64(3):155–160, 1997.

[4] E. Althaus and K. Mehlhorn. Maximum network flow with floating point arithmetic. *Information Processing Letters*, 66(3):109–113, 1998.

[5] A. Andersson, T. Hagerup, S. Nilsson, and R. Raman. Sorting in linear time? *Journal of Computer and System Sciences*, 57(1):74–93, Aug. 1998.

[6] S. Arikati, S. Chaudhuri, and C. Zaroliagis. All-pairs min-cut in sparse networks. *Journal of Algorithms*, 29:82–110, 1998.

[7] G. Barnes and J. A. Edmonds. Time-space lower bounds for directed st-connectivity on graph automata models. *SIAM Journal on Computing*, 27(4):1190–1202, August 1998.

[8] H. L. Bodlaender, J. S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, and Z. Tuza. Ranking of graphs. *SIAM Journal on Computing*, 11(1):168–181, February 1999.

[9] P. G. Bradford, G. J. Rawlins, and G. E. Shannon. Efficient matrix chain ordering in polylog time. *SIAM Journal on Computing*, 27(2):466–490, 1998.

[10] F. J. Brandenburg, M. Jünger, and P. Mutzel. Algorithmen zum automatischen Zeichnen von Graphen. *Informatik Spektrum*, 20(4):199–207, 1997.

[11] D. Breslauer, T. Jiang, and Z. Jiang. Rotations of periodic strings and short superstrings. *Journal of Algorithms*, 24(1):340–353, 1997.

[12] G. S. Brodal, J. L. Träff, and C. Zaroliagis. A parallel priority queue with constant time operations. *Journal of Parallel and Distributed Computing, Special Issue on Parallel Data Structures*, 49(1):4–21, 1998.

[13] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. *International Journal of Computational Geometry and Applications — special issue.* To appear.

[14] S. Chaudhuri, N. Garg, and R. Ravi. The p-neighbor k-center problem. *Information Processing Letters*, 65:131–134, 1998.

[15] S. Chaudhuri, K. V. Subrahmanyam, F. Wagner, and C. Zaroliagis. Computing mimicking networks. *Algorithmica, Special Issue on Graph Algorithms and Applications.* To appear.

[16] J. Cheriyan and K. Mehlhorn. An analysis of the highest-level selection rule in the preflow-push max-flow algorithm. *Information Processing Letters*, 69(5):239–242, 1999.

[17] K. W. Chong and T. W. Lam. Approximating biconnectivity in parallel. *Algorithmica*, 21:395–410, 1998.

[18] T. Christof, M. Jünger, J. Kececioglu, P. Mutzel, and G. Reinelt. A branch-and-cut approach to physical mapping of chromosomes by unique end-probes. *Journal of Computational Biology*, 4(4):433–447, 1997.

[19] C. Cooper, A. Frieze, K. Mehlhorn, and V. Priebe. Average-case complexity of shortest-paths problems in the vertex-potential model. *Random Structures Algorithms*. To appear.

[20] M. Crochemore, Z. Galil, L. Gasieniec, K. Park, and W. Rytter. Constant-time randomized parallel string matching. *SIAM Journal on Computing*, 26(4):950–960, 1997.

[21] M. Crochemore, L. Gasieniec, R. Hariharan, S. Muthukrishnan, and W. Rytter. A constant time optimal parallel algorithm for two-dimensional pattern matching. *SIAM Journal on Computing*, 27(3):668–681, June 1998.

[22] G. Das, S. Kapoor, and M. Smid. On the complexity of approximating euclidean traveling salesman tours and minimum spanning trees. *Algorithmica*, 19:447–460, 1997.

[23] M. de Berg, M. van Kreveld, and S. Schirra. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and Geographic Information Systems*, 25(4):243–257, 1998.

[24] X. Deng and S. Mahajan. The cost of derandomization: Computability or competitiveness. *SIAM Journal on Computing*, 26(3):786–802, 1997.

[25] K. Diks and T. Hagerup. More general parallel tree contraction: Register allocation and broadcasting in a tree. *Theoretical Computer Science*, 203(1):3–29, 1998.

[26] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(5):99–124, August 1998.

[27] P. Eades and P. Mutzel. Graph drawing algorithms. In M. J. Atallah, editor, *Algorithms and Theory of Computation Handbook*, chapter 9, pages 9–26. CRC Press, New York, USA, November 1998.

[28] H. Edelsbrunner and E. A. Ramos. Inclusion-exclusion complexes for pseudo-disk collections. *Discrete and Computational Geometry*, 17:287–306, 1997.

[29] P. Ferragina and R. Gross. Optimal on-line search and sublinear time update in string matching. *SIAM Journal on Computing*, 27(3):713–736, June 1998.

[30] P. Ferragina and R. Grossi. The string B-tree: A new data structure for string search in external memory and its applications. *Journal of the ACM)*, 1999. To appear.

[31] R. Fleischer. On the Bahncard problem. *Theoretical Computer Science*. To appear.

[32] R. Fleischer and C. Hirsch. Applications of graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*. Teubner-Verlag, Leipzig, 1999. In preparation.

[33] F. Follert, E. Schömer, J. Sellen, M. Smid, and C. Thiel. Computing the largest empty anchored cylinder, and related problems. *International Journal of Computational Geometry and Applications*, 7(6):563–580, 1997.

[34] G. N. Frederickson and R. Solis-Oba. Algorithms for measuring perturbability in matroid optimization. *Combinatorica*, 18(4):503–518, 1998.

[35] L. Gasieniec and A. Pelc. Broadcasting with a bounded fraction of faulty nodes. *Journal of Parallel and Distributed Computing*, 42(1):11–20, 1997.

[36] M. J. Golin, R. Raman, C. Schwarz, and M. Smid. Randomized data structures for the dynamic closest-pair problem. *SIAM Journal on Computing*, 27(4):1036–1072, August 1998.

[37] A. Gupta and S. Mahajan. Using amplification to compute majority with small majority gates. *Computational Complexity*, 6(1):46–63, 1997.

[38] P. Gupta, R. Janardan, and M. Smid. Efficient algorithms for counting and reporting pairwise intersections between convex polygons. *Information Processing Letters*, 69(1):7–13, January 1999.

[39] P. Gupta, R. Janardan, M. Smid, and B. Dasgupta. The rectangle enclosure and point-dominance problems revisited. *International Journal of Computational Geometry and Applications*, 7(5):437–456, 1997.

[40] T. Hagerup. Allocating independent tasks to parallel processors: An experimental study. *Journal of Parallel and Distributed Computing*, 47(2):185–197, 1997.

[41] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde. Characterizing multiterminal flow networks and computing flows in networks of small treewidth. *Journal of Computer and System Sciences*, 57(3):366–375, Dec. 1998.

[42] S. Hert and V. Lumelsky. Motion planning in R^3 for multiple tethered robots. *IEEE Transactions on Robotics and Automation*. To appear.

[43] C. Hundack, H. J. Prömel, and A. Steger. Extremal graph problems for graphs with a color-critical vertex. In B. Bellobás and A. Thomason, editors, *Combinatorics, Geometry and Probability: a tribute to Paul Erdös*, pages 421–433. Cambridge University Press, Cambridge, 1997.

[44] M. Jünger, S. Leipert, and P. Mutzel. A note on computing a maximal planar subgraph using PQ-trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(7):609–612, 1998.

[45] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algori thms. *Journal of Graph Algorithms and Applications (JGAA)* ***http://www.cs.brown.edu/publications/jgaa/***, 1(1):1–25, 1997.

[46] M. Jünger, P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of a minor-excluded class of graphs. *Discrete Mathematics*, 182:169–176, 1998.

[47] M. Kaufmann, U. Meyer, and J. F. Sibeyn. Matrix transpose on meshes: Theory and practice. *Computers and Artificial Intelligence*, 16(2):107–140, May 1997.

[48] M. Kaufmann, R. Raman, and J. F. Sibeyn. Routing on meshes with buses. *Algorithmica*, 18(3):417–444, 1997.

[49] J. D. Kececioglu, H.-P. Lenhof, K. Mehlhorn, P. Mutzel, K. Reinert, and M. Vingron. A polyhedral approach to sequence alignment problems. *Discrete Applied Mathematics*. To appear.

[50] C. W. Keßler and J. L. Träff. Language and library support for practical PRAM programming. *Parallel Computing*, 1998. To appear.

[51] L. Khachiyan and L. Porkolab. Integer optimization on convex semi-algebraic sets. *Discrete and Computational Geometry*, 1999. To appear.

[52] O. Kohlbacher and H.-P. Lenhof. New developments in protein-protein docking. *Macromol. Theory and Simulation*, 1999. To appear.

[53] P. Krysta and L. Pacholski. The STO problem is NP-complete. *Journal of Symbolic Computation*, 27(2):207–219, February 1999.

[54] H.-P. Lenhof, B. Morgenstern, and K. Reinert. An exact solution for the segment-to-segment multiple sequence alignment problem. *BIOINFORMATICS*. To appear.

[55] H.-P. Lenhof, K. Reinert, and M. Vingron. A polyhedral approach to RNA sequence structure alignment. *Journal of Computational Biology*, 5(3):517–530, 1998.

[56] S. Leonardi. On-line network routing. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 242–267. LNCS 1442, Springer Verlag, 1998.

[57] K. Mehlhorn, M. Müller, S. Näher, S. Schirra, M. Seel, C. Uhrig, and J. Ziegler. A computational basis for higher-dimensional computational geometry and applications. *Computational Geometry: Theory and Applications*, 10(4):289–304, July 1998.

[58] P. Mutzel. An alternative method to crossing minimization on hierarchical graphs. *SIAM Journal on Optimization*, 1999. To appear.

[59] P. Mutzel. Optimization in leveled graphs. In P. Pardalos and C. Floudas, editors, *Encyclopedia of Optimization*. Kluwer Academic Publishers, 1999. To appear.

[60] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs and Combinatorics*, 14:59–73, 1998.

[61] A. Panconesi, M. Papatriantafilou, P. Tsigas, and P. Vitanyi. Randomized naming using wait-free shared variables. *Distributed Computing*, 11(3):113–124, 1998.

[62] M. Papatriantafilou and P. Tsigas. Self-stabilizing wait-free clock synchronization. *Parallel Processing Letters*, 7(3):321–328, 1997.

[63] E. A. Ramos. Intersection of unit-balls and diameter of a point set in R^3. *Computational Geometry Theory and Applications*, 8:57–65, 1997.

[64] U. Rathe, P. Sanders, and P. Knight. A case study in scalability: an ADI method for the two-dimensional time-dependent Dirac equation. *Parallel Computing*, 1999. To appear.

[65] P. Sanders. Random permutations on distributed, external and hierarchical memory. *Information Processing Letters*, 67(6):305–309, 1998.

[66] P. Sanders. Randomized priority queues for fast parallel access. *Journal Parallel and Distributed Computing*, 49(1):86–97, 1998.

[67] J. F. Sibeyn. Routing on triangles, tori and honeycombs. *International Journal on the Foundations of Computer Science*, 8(3):269–287, 1997.

[68] J. F. Sibeyn. List ranking on meshes. *Acta Informatica*, 35(7):543–566, 1998.

[69] J. F. Sibeyn, M. Grammatikakis, D. Hsu, and M. Kraetzl. Packet routing in fixed-connection networks: A survey. *Journal of Parallel and Distributed Computing*, 54:77–132, 1998.

[70] J. F. Sibeyn, F. Guillaume, and T. Seidel. Practical parallel list ranking. *Journal of Parallel and Distributed Computing*, 56(2):156–180, February 1999.

[71] J. F. Sibeyn, B. Juurlink, and P. S. Rao. Gossiping on meshes and tori. *IEEE Transactions on Parallel and Distributed Systems*, 9(6):513–525, June 1998.

[72] J. F. Sibeyn and M. Kaufmann. Randomized multipacket routing and sorting on meshes. *Algorithmica*, 17:224–244, 1997.

[73] J. F. Sibeyn, M. Kaufmann, and B. S. Chlebus. Deterministic permutation routing on meshes. *Journal of Algorithms*, 22(1):111–141, 1997.

[74] K. Sridharan, C. R. Subramanian, and N. Sudha. Some properties of touching distances for polygons and polyhedra. *Applied Mathematics Letters*, 11(5):1–7, August 1998.

[75] C. R. Subramanian, M. Fürer, and C. E. Veni Madhavan. Algorithms for coloring semi-random graphs. *Random Structures & Algorithms*, 13(2):125–158, September 1998.

In Conference Proceedings

[1] P. Agarwal, L. Arge, G. S. Brodal, and J. Vitter. I/O-efficient dynamic point location in monotone subdivisions. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 11–20, 1999.

[2] S. Albers, S. Arora, and S. Khanna. Page replacement for general caching problems. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 31–40, 1999.

[3] S. Albers, M. Charikar, and M. Mitzenmacher. On delayed information and action in on-line algorithms. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 71–80, 1998.

[4] S. Albers, N. Garg, and S. Leonardi. Minimizing stall time in single and parallel disk systems. In *Proceedings 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, pages 454–462, 1998.

[5] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 842–843, Baltimore, USA, 1999.

[6] S. Albers and M. Mitzenmacher. Average case analyses of first-fit and random-fit bin packing. In *Proceedings 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 290–299, 1998.

[7] S. Albers and G. Schmidt. Scheduling with unexpected machine breakdowns. In *Proceedings of APPROX'99*. LNCS, Springer Verlag, 1999. To appear.

[8] D. Alberts, G. Cattaneo, G. Italiano, U. Nanni, and C. Zaroliagis. A software library of dynamic graph algorithms. In *Proc. Workshop on Algorithms and Experiments – ALEX'98*, pages 129–136, 1998.

[9] D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. The design of the AGD-Algorithms Library. In G. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE '97)*, 1997. Venice, Italy, Sept. 11-13, http://www.dsi.unive.it/∼wae97/.

[10] P. Alefragis, C. Goumopoulos, E. Housos, P. Sanders, T. Takkula, and D. Wedelin. Parallel crew scheduling in PAROS. In *4th Euro-Par*, number 1470 in LNCS, pages 1104–1113. Springer, 1998.

[11] H. Bast. Dynamic scheduling with incomplete information. In *Proceedings of the 10th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA-98)*, pages 182–191, Puerto Vallarta, Mexico, June 1998. Association for Computing Machinery (ACM), ACM Press.

[12] T. Biedl, M. Kaufmann, and P. Mutzel. Drawing planar partitions II: HH-drawings. In J. Hromkovic and O. Sykora, editors, *Proceedings of the 24th Workshop on Graph-Theoretic Concepts in Computer Science (WG-98)*, volume 1517 of *Lecture Notes in Computer Science*, pages 124–136, Smolenice, 1998. Springer.

[13] H. L. Bodlaender and T. Hagerup. Tree decompositions of small diameter. In *Proceedings of the 23th International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 702–712. Springer, Berlin, 1998.

[14] F. J. Brandenburg, M. Jünger, P. Mutzel, and T. Lengauer. SPP 731 : Algorithmen zum automatischen Zeichnen von Graphen im Rahmen des DFG-Schwerpunkts "Effiziente Algorithmen für diskrete Probleme und ihre Anwendungen". In M. Jarke, K. Pasedach, and K. Pohl, editors, *Informatik '97: Informatik als Innovationsmotor: 27. Jahrestagung der Gesellschaft für Informatik*, pages 58–67, Aachen, 1997. Springer.

[15] G. S. Brodal. Predecessor queries in dynamic integer sets. In R. Reischuk and M. Morvan, editors, *Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS-97)*, volume 1200 of *Lecture Notes in Computer Science*, pages 21–32, Lübeck, Germany, 1997. Springer.

[16] G. S. Brodal. Finger search trees with constant update time. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, pages 540–549, San Francisko, USA, 1998. Association of Computing Machinery/Society for Industrial and Applied Mathematics, ACM Press / SIAM.

[17] G. S. Brodal and J. Katajainen. Worst-case efficient external-memory priority queues. In S. Arnborg and L. Ivansson, editors, *Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (SWAT-98)*, volume 1432 of *Lecture Notes in Computer Science*, pages 107–118, Stockholm, Sweden, July 1998. Springer.

[18] G. S. Brodal and M. C. Pinotti. Comparator networks for binary heap construction. In S. Arnborg and L. Ivansson, editors, *Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (SWAT-98)*, volume 1432 of *Lecture Notes in Computer Science*, pages 158–168, Stockholm, Sweden, July 1998. Springer.

[19] H. Brönnimann, C. Burnikel, and S. Pion. Interval analysis yields efficient dynamic filters for computational geometry. In *Proceedings of the 14th International Annual ACM Symposium on Computational Geometry (SCG-98)*, pages 165–174, Minneapolis, Minnesota, June 1998. Association Computing Machinery (ACM), ACM Press.

[20] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array (QUASAR). In *Proc. of the Third Annual International Conference on Computational Molecular Biology (RECOMB 99)*, pages 77–83, 1999.

[21] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Efficient exact geometric computation made easy. In *Proceedings of 15th ACM Symposium on Computational Geometry*, 1999. To appear.

[22] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. In *Proceedings of the 14th International Annual ACM Symposium on Computational Geometry (SCG-98)*, pages 175–183, Minneapolis, USA, 1998. Association of Computing Machinery (ACM), ACM Press.

[23] S. Chaudhuri, K. V. Subrahmanyam, F. Wagner, and C. Zaroliagis. Computing mimicking networks. In K. G. Larsen, S. Skyum, and G. Winskel, editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP-98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 556–567, Aalborg, Denmark, 1998. Springer.

[24] J. Cheriyan, T. Jordán, and Z. Nutov. Approximating k-outconnected subgraph problems. In *Proceedings 1st Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)*, Lecture Notes in Computer Science, pages 77–88. LNCS 1444, Springer Verlag, 1998.

[25] B. S. Chlebus, A. Czumaj, and J. F. Sibeyn. Routing on the PADAM: Degrees of optimality. In C. Lengauer, M. Griebl, and S. Gorlatch, editors, *Proceedings of the 3rd International Euro-Par Conference on Parallel Processing (Euro-Par-97)*, volume 1300 of *Lecture Notes in Computer Science*, pages 272–279, Passau, Germany, 1997. Springer.

[26] K. W. Chong, Y. Han, and T. Lam. On the parallel time complexity of undirected connectivity and minimum spanning trees. In *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 225–234, 1999.

[27] K. W. Chong and E. A. Ramos. Improved deterministic parallel padded sorting. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium on Algorithms (ESA-98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 405–416, Venice, Italy, 1998. Springer.

[28] A. Crauser and P. Ferragina. On constructing suffix arrays in external memory. In *European Symposium on Algorithms (ESA)*, 1999. To appear.

[29] A. Crauser, P. Ferragina, K. Mehlhorn, U. Meyer, and E. A. Ramos. Randomized external-memory algorithms for some geometric problems. In *Proceedings of the 14th International Annual ACM Symposium on Computational Geometry (SCG-98)*, pages 259–268, Minneapolis, Minnesota, 1998. ACM, ACM Press.

[30] A. Crauser, K. Mehlhorn, and U. Meyer. Kürzeste-Wege-Berechnung bei sehr großen Datenmengen. In O. Spaniol, editor, *Promotion tut not: Innovationsmotor "Graduiertenkolleg"*, volume 21 of *Aachener Beiträge zur Informatik*, pages 113–132, Aachen, Germany, September 1997. Augustinus.

[31] A. Crauser, K. Mehlhorn, U. Meyer, and P. Sanders. A parallelization of Dijkstra's shortest path algorithm. In L. Brim, J. Gruska, and J. Zlatuska, editors, *Proceedings of the 23rd International Symposium on the Mathematical Foundations of Computer Science (MFCS-98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 722–731, Brno, Czech Republic, August 1998. Springer.

[32] A. Czumaj, P. Ferragina, L. Gasieniec, S. Muthukrishnan, and J. L. Träff. The architecture of a software library for string processing. In G. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE-97)*, pages 166–176, Venice, Italy, 1997. Universitá Ca' Foscari di Venezia.

[33] G. Das, D. Gunopulos, R. Fleischer, L. Gasieniec, and J. Kärkkäinen. Episode matching. In A. Apostolico and J. Hein, editors, *Proceedings of the 8th Symposium on Combinatorial Pattern Matching (CPM-97)*, volume 1264 of *Lecture Notes in Computer Science*, pages 12–27, Aarhus, Denmark, June 1997. Springer.

[34] T. K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *Proc. 10th. SIAM Sympos. Discr. Algorithms*, pages 893–894, 1999.

[35] T. K. Dey, K. Mehlhorn, and E. A. Ramos. Curve reconstruction: Connecting dots with good reason. In *Proc. 15th Ann. Sympos. Comput. Geom.*, 1999. To appear.

[36] Y. Dinitz, N. Garg, and M. Goemans. On the single-source unsplittable flow problem. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 290–299, 1998.

[37] P. Eades, J. Marks, P. Mutzel, and S. North. Graph drawing contest report. In S. H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD-98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 423–435, Montreal, Canada, 1998. Springer.

[38] T. Erlebach and K. Jansen. Efficient implementation of an optimal greedy algorithm for wavelength assignment. In *Proceedings of Workshop on Algorithm Engineering (WAE'98)*, pages 13 – 24, 1998.

[39] T. Erlebach and K. Jansen. Maximizing the number of connections in optical tree networks. In *Proceedings International Symposium on Algorithms and Computation (ISAAC'98)*, pages 179 – 188. LNCS 1533, Springer Verlag, 1998.

[40] M. Farach, P. Ferragina, and S. Muthukrishnan. Overcoming the memory bottleneck in suffix tree construction. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 174–183, 1998.

[41] P. Ferragina and F. Luccio. Multi-string search in BSP. In *SEQS: Compression and Complexity of Sequences 1997*. IEEE Press, 1998.

[42] S. Fialko and P. Mutzel. A new approximation algorithm for the planar augmentation problem. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-98)*, pages 260–269, San Francisco, USA, 1998. ACM, ACM Press / SIAM.

[43] R. Fleischer. FUN with implementing algorithms. In *Proceedings of* FUN with Algorithms*, Elba, Italy*. Carleton Scientific, 1998.

[44] R. Fleischer. On the Bahncard problem. In W.-L. Hsu and M.-Y. Kao, editors, *Proceedings of the 4th International Conference on Computing and Combinatorics (COCOON-98)*, volume 1449 of *Lecture Notes in Computer Science*, pages 65–74, Taipei, Taiwan, R.o.C., 1998. Springer.

[45] R. Fleischer and S. S. Seiden. Page replication—Variations on a theme. In *International Conference on Optimization (SIGOPT'99), Trier, Germany*, 1999.

[46] G. Frederickson and R. Solis-Oba. Rooted spanning trees with small weight and average length. In *Proceedings 6th Italian Conference on Theoretical Computer Science*, pages 114–125. World Scientific Publishing, 1998.

[47] D. Frigioni, T. Miller, U. Nanni, G. Pasqualone, G. Schäfer, and C. Zaroliagis. An experimental study of dynamic algorithms for directed graphs. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium on Algorithms (ESA-98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 368–380, Venice, Italy, 1998. Springer.

[48] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 300–309, 1998.

[49] L. Gasieniec and P. Indyk. Efficient parallel computing with memory faults. In B. S. Chlebus and L. Czaja, editors, *Proceedings of the 11th Symposium on Fundamentals of Computation Theory (FCT-97)*, volume 1279 of *Lecture Notes in Computer Science*, pages 188–197, Krakcw, Poland, 1997. Springer.

[50] L. Gasieniec, J. Jansson, A. Lingas, and A. Oestlin. On the complexity of computing evolutionary trees. In T. Jiang and D. T. Lee, editors, *Proceedings of the 3rd International Conference on Computing and Combinatorics (COCOON-97)*, volume 1276 of *Lecture Notes in Computer Science*, pages 134–145, Shanghai, China, 1997. Springer.

[51] J. Gergov. Algorithms for compile-time memory optimization. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 907–908, January 1999.

[52] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In S. H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD-98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 167–182, Montreal, Canada, 1998. Springer.

[53] T. Hagerup. Simpler and faster dictionaries on the AC^0 RAM. In K. G. Larsen and S. Skyum, editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP-98)*, volume 1443 of *Lecture Notes in Computer Science*, pages 79–90, Aalborg, Denmark, July 1998. Springer.

[54] T. Hagerup. Sorting and searching on the Word RAM. In M. Morvan, C. Meinel, and D. Krob, editors, *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS-98)*, volume 1373 of *Lecture Notes in Computer Science*, pages 366–398, Paris, France, 1998. Springer.

[55] T. Hagerup, P. Sanders, and J. L. Träff. An implementation of the binary blocking flow algorithm. In K. Mehlhorn, editor, *Proceedings of the 2nd Workshop on Algorithm Engineering (WAE-98)*, pages 143–154, Saarbrücken, Germany, August 1998. Max-Planck-Institut für Informatik.

[56] M. Henzinger and S. Leonardi. Scheduling multicasts on unit capacity trees and meshes. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 438–447, 1999.

[57] C. Hundack, P. Mutzel, I. Pouchkarev, and S. Thome. Arche: A graph drawing system for archaeology. In G. Di Battista, editor, *Proceedings of the 5th Symposium on Graph Drawing (GD-97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 297–302, Rome, Italy, 1997. Springer.

[58] K. Jansen. An approximation algorithm for bin packing with conflicts. In *Proceedings Skandinavian Workshop on Algorithm Theory (SWAT'98)*, pages 35 – 46. LNCS 1432, Springer Verlag, 1998.

[59] K. Jansen. The mutual exclusion scheduling problem for permutation and comparability graphs. In M. Morvan, C. Meinel, and D. Krob, editors, *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS-98)*, volume 1373 of *Lecture Notes in Computer Science*, pages 287–297, Paris, France, 1998. Springer.

[60] K. Jansen and L. Porkolab. Improved approximation schemes for scheduling unrelated parallel machines. In *Proceedings 31st Annual ACM Symposium on Theory of Computing (STOC'99)*, 1999. To appear.

[61] K. Jansen and L. Porkolab. Linear-time approximation schemes for scheduling malleable parallel tasks. In *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 490–498, 1999.

[62] K. Jansen, R. Solis-Oba, and M. Sviridenko. Makespan minimization in job shops: a polynomial time approximation scheme. In *Proceedings 31st Annual ACM Symposium on Theory of Computing (STOC'99)*, 1999. To appear.

[63] B. Jung, H.-P. Lenhof, P. Müller, and C. Rüb. Simulating synthetic polymer chains in parallel. In *Proc. of the 7th International Conference on High Performance Computing (HPCN), Amsterdam, LNCS 1593*, pages 13–22. Springer, April 1999.

[64] M. Jünger, E. K. Lee, P. Mutzel, and T. Odenthal. A polyhedral approach to the multi-layer crossing minimization problem. In G. Di Battista, editor, *Proceedings of the 5th Symposium on Graph Drawing (GD-97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 13–24, Rome, Italy, 1997. Springer.

[65] M. Jünger, S. Leipert, and P. Mutzel. Pitfalls of using PQ-trees in automatic graph drawing. In G. Di Battista, editor, *Proceedings of the 5th Symposium on Graph Drawing (GD-97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204, Rome, Italy, 1997. Springer.

[66] M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In S. H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD-98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237, Montreal, Canada, 1998. Springer.

[67] L. Khachiyan and L. Porkolab. Computing integral points in convex semi-algebraic sets. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 162–171, 1997.

[68] G. W. Klau and P. Mutzel. Optimal compaction of orthogonal grid drawings. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

[69] P. Krysta and K. Loryś. Efficient approximation algorithms for the achromatic number. In *Proceedings 7th Annual European Symposium on Algorithms (ESA'99)*. LNCS, Springer Verlag, Berlin, 1999. To appear.

[70] P. Krysta and R. Solis-Oba. Approximation algorithms for bounded facility location. In *Proceedings 5th Annual International Computing and Combinatorics Conference (COCOON'99)*. LNCS, Springer Verlag, Berlin, 1999. To appear.

[71] H.-P. Lenhof, K. Reinert, and M. Vingron. A polyhedral approach to RNA sequence structure alignment. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB 98)*, pages 153–162, 1998.

[72] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosén. On-line randomized call-control revisited. In *Proceedings 9th ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, pages 323–332, 1998.

[73] S. Mahajan, E. A. Ramos, and K. Subrahmanyam. Solving some discrepancy problems in NC. In S. Ramesh and G. Sivakumar, editors, *Proceedings of the 17th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS-97)*, volume 1346 of *Lecture Notes in Computer Science*, pages 22–36, Kharagpur, India, 1997. Springer.

[74] L. Malmi and E. Soisalon-Soininen. Group updates for relaxed height-balanced trees. In *Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1999. To appear.

[75] K. Mehlhorn, M. Müller, S. Näher, S. Schirra, M. Seel, C. Uhrig, and J. Ziegler. A computational basis for higher-dimensional computational geometry and applications. In *13th Symposium on Computational Geometry (SCG-97): Nice, France, June 4-6, 1997; proceedings*, pages 254–263S., New York, NY, 1997. ACM Press.

[76] K. Mehlhorn and S. Näher. From algorithms to working programs: On the use of program checking in LEDA. In L. Brim, J. Gruska, and J. Zlatuska, editors, *Proceedings of the 23rd International Symposium On Mathematical Foundations of Computer Science (MFCS-98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 84–93, Brno, Czech Republic, August 1999. Springer.

[77] K. Mehlhorn, S. Näher, and C. Uhrig. The LEDA platform for combinatorial and geometric computing. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings of the 24th International Colloquium on Automata, Languages, and Programming (ICALP-97)*, volume 1256 of *Lecture Notes in Computer Science*, pages 7–16, Bologna, Italy, 1997. Springer.

[78] U. Meyer and P. Sanders. δ-stepping: A parallel single source shortest path algorithm. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium on Algorithms (ESA-98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 393–404, Venice, Italy, August 1998. Springer.

[79] U. Meyer and J. F. Sibeyn. Gossiping large packets on full-port tori. In D. Pritchard and J. Reeve, editors, *Proceedings of the 4th International Euro-Par Conference (Euro-Par-98)*, volume 1470 of *Lecture Notes in Computer Science*, pages 1040–1046, Southampton, United Kingdom, September 1998. Springer.

[80] T. Miller and C. Zaroliagis. A first experimental study of a dynamic transitive closure algorithm. In G. Italiano and S. Orlando, editors, *Proceedings of the Workshop on Algorithm Engineering (WAE-97)*, pages 64–73, Venice, Italy, 1997. Universitá Ca' Foscari di Venezia.

[81] P. Mutzel, C. Gutwenger, R. Brockenauer, S. Fialko, G. Klau, M. Krüger, T. Ziegler, S. Näher, D. Alberts, D. Ambras, G. Koch, M. Jünger, C. Buchheim, and S. Leipert. AGD: A Library of Algorithms for Graph Drawing (poster-abstract). In S. Whitesides, editor, *Graph Drawing (Proc. GD '98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 456–457. Springer-Verlag, 1998.

[82] P. Mutzel and R. Weiskircher. Two-layer planarization in graph drawing. In K.-Y. Chwa and O. H. Ibarra, editors, *Proceedings of the 9th International Symposium on Algorithms and Computation (ISAAC-98)*, volume 1533 of *Lecture Notes in Computer Science*, pages 69–78, Taejon, Korea, December 1998. Korea Information Science Society; Korea Advanced Institute of Science and Technology, Springer.

[83] P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph. In G. P. Cornuejols, editor, *Integer Programming and Combinatorial Optimization (IPCO '99), Proceedings of the Seventh Conference*, volume 1610 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. To appear.

[84] P. Mutzel and T. Ziegler. The constrained crossing minimization problem - a first approach. In P. Kall and H.-J. Lthi, editors, *Operations Research Proceedings 1998*, pages 125–134, Zürich, Switzerland, 1999. Springer.

[85] D. Neumann, E. Haltner, C.-M. Lehr, O. Kohlbacher, and H.-P. Lenhof. Investigating the sugar-lectin interaction by computational chemistry: tunneling the epithelial barrier. In *Abstracts of the AAPS Annual Meeting, San Francisco, USA*. American Association of Pharmazeutical Scientists, Nov. 1998.

[86] Z. Nutov. Approximating multiroot 3-outconnected subgraphs. In *Proceedings 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 951–952, 1999.

[87] E. A. Ramos. On range reporting, ray shooting and k-level construction. In *Proc. 15th Annu. Sympos. Comput. Geom.*, 1999. To appear.

[88] C. Rüb. On Batcher's merge sorts as parallel sorting algorithms. In M. Morvan, C. Meinel, and D. Krob, editors, *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS-98)*, volume 1373 of *Lecture Notes in Computer Science*, pages 410–420, Paris, France, 1998. Springer.

[89] P. Sanders. Tree shaped computations as a model for parallel applications. In A. Bode, editor, *Anwendungsbezogene Lastverteilung (ALV)*, pages 123–132, München, 1998. SFB 342, TU München. Appeared as Technical Report TUM-I9806 SFB-Bericht Nr. 342/01/98 A.

[90] P. Sanders. Accessing multiple sequences through set associative caches. In *ICALP*, LNCS, 1999. To appear.

[91] P. Sanders. Fast priority queues for cached memory. In *Workshop in Algorithmic Engineering and Experimentation (ALENEX)*, Lecture Notes in Computer Science. Springer, 1999.

[92] P. Sanders. Random permutations on distributed, external and hierarchical memory. In *PARS-Mitteilungen*, number 17 in PARS-Mitteilungen, pages 160–165, Karlsruhe, 1999. GI-ITG: Parallel-Algorithmen, -Rechnerstrukturen und -Systemsoftware.

[93] P. Sanders, R. Reussner, L. Prechelt, and M. Müller. SkaMPI: A detailed, accurate MPI benchmark. In V. Alexandrov and J. Dongarra, editors, *Proceedings of the 5th European PVM/MPI Users' Group Meeting*, volume 1497 of *Lecture Notes in Computer Science*, pages 52–59, Liverpool, UK, September 1998. Springer.

[94] P. Sanders, T. Takkula, and D. Wedelin. High performance integer optimization for crew scheduling. In *7th International Conference on High Performance Computing and Networking Europe*, number 1593 in LNCS, pages 3–12, 1999. To appear.

[95] S. Schirra. A case study on the cost of geometric computing. In *Proc. of ALENEX'99*, 1999. To appear.

[96] C. P. Schnorr and C. R. Subramanian. Almost optimal (on the average) algorithms for boolean matrix product witnesses, computing the diameter. In M. Luby, J. Rolim, and M. Serna, editors, *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-98)*, volume 1518 of *Lecture Notes in Computer Science*, pages 218–231, Barcelona, Spain, October 1998. Springer.

[97] S. S. Seiden. A manifesto for the computational method. In *Proceedings of* FUN with Algorithms, *Elba, Italy*. Carleton Scientific, 1998.

[98] J. F. Sibeyn. Routing with finite speeds of memory and network. In I. Prívara and P. Ruzicka, editors, *Proceedings of the 22nd Symposium on the Mathematical Foundations of Computer Science (MFCS-97)*, volume 1295 of *Lecture Notes in Computer Science*, pages 488–497, Bratislava, Slovakia, 1997. Springer.

[99] J. F. Sibeyn. Sample sort on meshes. In C. Lengauer, M. Griebl, and S. Gorlatch, editors, *Proceedings of the 3rd International Euro-Par Conference on Parallel Processing (Euro-Par-97)*, volume 1300 of *Lecture Notes in Computer Science*, pages 389–398, Passau, Germany, 1997. Springer.

[100] J. F. Sibeyn. Solving fundamental problems on sparse-meshes. In S. Arnborg and L. Ivansson, editors, *Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (SWAT-98)*, volume 1432 of *Lecture Notes in Computer Science*, pages 288–299, Stockholm, July 1998. Springer.

[101] J. F. Sibeyn. Better deterministic routing on meshes. In *Proc. 13th International Parallel Processing Symposium*. IEEE Computer Society Press, 1999.

[102] J. F. Sibeyn. External selection. In *Proc. 16th Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 291–301. Springer, 1999.

[103] J. F. Sibeyn, F. Guillaume, and T. Seidel. Practical parallel list ranking. In G. Bilardi, A. Ferreira, R. Lüling, and J. Rolim, editors, *Proceedings of the 4th Symposium on Solving Irregularly Structured Problems in Parallel (IRREGULAR-97)*, volume 1253 of *Lecture Notes in Computer Science*, pages 25–36, Paderborn, Germany, 1997. Springer.

[104] R. Solis-Oba. 2-approximation algorithm for finding a spanning tree with maximum number of leaves. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium on Algorithms (ESA-98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 441–452, Venice, Italy, August 1998. Springer.

[105] R. Solis-Oba and G. N. Frederickson. Rooted spanning trees with small weight and average length. In P. Degano, U. Vaccaro, and G. Pirillo, editors, *Theoretical Computer Science*, pages 114–125, Prato, Italy, 1998. IC-EACTS, World Scientific.

[106] C. R. Subramanian. A generalization of Janson inequalities and its application to finding shortest paths. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 795–804, 1999.

Part IV

The Programming Logics Group

1 Personnel

Director:
Harald Ganzinger

Senior research scientist:
Andreas Podelski

Researchers:
Alexander Bockmayr (–October 1998)
Witold Charatonik
Manfred Jaeger (October 1997–)
Seán Matthews (–February 1999)
Hans de Nivelle (May 1999–)
Andreas Nonnengart (–March 1999)
Sergei Vorobyov (–March 1999)
Uwe Waldmann
Christoph Weidenbach
Emil Weydert

Post-doctoral fellows and long-term guests:
Richard Booth (January 1999–; previously at the University of Manchester)
Georgio Delzanno (October 1996–; previously at University of Genova)
Florent Jacquemard (January 1996–September 1998; previously at the LRI, Orsay)
Ralamboniaina Ramangalahy (January 1996–April 1998; previously at the EDF, Clamart)
Viorica Sofronie-Stokkermans (October 1997–; previously at RISC, Linz)
Jean-Marc Talbot (October 1998–; previously at LIFL, Lille)
Leon van der Torre (March 1997–November 1997; December 1998–April 1999; previously at the Erasmus University Rotterdam)
Margus Veanes (September 1997–; previously at the University of Uppsala)
Jinzhao Wu (January 1998–; previously at the Texas A&M University)

Ph.D. students:
Hubert Baumeister (–November 1998)
Fritz Eisenbrand
Jörn Hopf
Ullrich Hustadt (–November 1997)
Thomas Kasper (–April 1999)
Patrick Maier (July 1998–)
Christoph Meyer
Supratik Mukhopadhyay (August 1997–)
Renate Schmidt (–December 1997)
Georg Struth (–November 1997)
Jürgen Stuber
Miroslava Tzakova
Luca Viganò (–October 1997)

Secretaries:
Brigitta Hansen (January 1999–)
Ellen Fries (–November 1998)
Christine Kiesel

2 Visitors

1997

Francois Fages	21.04.97–23.04.97	E.N.S. Paris
Leo Bachmair	29.05.97–10.07.97	SUNY Stony Brook
Patrick Cousot	01.06.97–02.06.97	E.N.S. - DMI Paris
Laurent Perron	11.06.97–12.06.97	E.N.S. Paris
Pawel Rzechonek	12.06.97–31.08.97	University Wroclaw
Andrzej Lukaszewski	12.06.97–10.08.97	University Wroclaw
Andrei Voronkov	15.06.97–31.07.97	University Uppsala
Daniel Kucner	16.06.97–31.08.97	University Wroclaw
Grazyna Salbierz	16.06.97–31.08.97	University Wroclaw
Slawomir Ziolkowski	16.06.97–16.08.97	University Wroclaw
Mandana Eibegger	17.06.97–20.06.97	TU Wien
Hans Moser	17.06.97–20.06.97	TU Wien
Tobias Nipkow	19.06.97–20.06.97	TU München
Suman Roy	24.07.97–26.07.97	Institute of Science Bangalore
David Basin	31.08.97–01.09.97	Universität Freiburg
Tomasz Zajac	01.08.97–23.08.97	University Wroclaw
Lincoln Wallen	18.08.97–29.09.97	Oxford University
Andrei Voronkov	01.09.97–15.10.97	University Uppsala
Damian Niwinski	24.09.97–20.12.97	University of Warsaw
Andrea Formisano	06.10.97–06.02.98	Università dell'Aquila
Stefano Sorgi	06.10.97–06.02.98	Università dell'Aquila
Tomasz Charatonik	08.10.97–14.11.97	University Wroclaw
Hans de Nivelle	08.10.97–15.11.97	University of Amsterdam
Dexter Kozen	16.10.97–19.10.97	Cornell University
David McAllister	22.10.97–23.10.97	AT&T Labs
Moshe Y. Vardi	26.10.97–28.10.97	Rice University
Hans Jürgen Ohlbach	02.11.97–04.11.97	Imperial College, London
Yannis Dimopoulos	12.11.97–14.11.97	Universität Freiburg
Pierre Wolper	18.11.97	Université de Liege
Andrei Voronkov	20.11.97–21.11.97	University Uppsala
Leo Bachmair	26.11.97–27.11.97	SUNY Stony Brook
Hans van Maaren	09.12.97–11.12.97	University Delft
Damian Niwinski	07.01.98–29.01.98	University of Warsaw
John Gallagher	15.01.98–16.01.98	University of Bristol
Johannes Waldmann	26.01.98–27.01.98	Universität Jena
Luca Viganò	29.01.98–30.01.98	Universität Freiburg
Leon van der Torre	03.02.98–09.02.98	IRIT Toulouse
Javier Esparza	18.02.98–19.02.98	TU München
Manfred Schramm	25.02.98	FH Ravensburg-Weingarten

1998

Gerhard Schellhorn	04.03.98–05.03.98	Universität Ulm
Wolfgang Reif	04.03.98–05.03.98	Universität Ulm
Reiner Hähnle	06.03.98	Universität Karlsruhe
Luca Viganò	09.03.98–11.03.98	Universität Freiburg
Renate Schmidt	09.03.98–11.03.98	Manchester Metropolitan University
Ullrich Hustadt	09.03.98–11.03.98	Manchester Metropolitan University
Bernd Fischer	09.03.98–11.03.98	TU Braunschweig
Hans de Nivelle	15.04.98–06.05.98	University of Amsterdam
Michele Bugliesi	09.05.98–14.05.98	University Padova
Nevin Heintze	19.05.98–10.06.98	Bell Labs
Yannis Dimopoulos	31.05.98–19.06.98	University of Cyprus
Leo Bachmair	01.06.98–02.08.98	SUNY Stony Brook
Pablo Argon	15.06.98	Ecole Central Nantes
Paola Inverardi	15.06.98–18.06.98	University of L'Aquila
Monica Nesi	15.06.98–18.06.98	University of L'Aquila
Andrei Voronkov	03.07.98–31.08.98	Uppsala University
Véronique Cortier	06.07.98–01.08.98	ENS Cachan
Christopher Lynch	11.07.98–19.07.98	Clarkson University
Mateja Jamnik	13.07.98–15.07.98	University of Edinburgh
Sabine Glesner	15.07.98	Universität Karlsruhe
Richard Booth	03.08.98–09.08.98	University of Manchester
M.J. Gabbay	14.08.98–92.09.98	Trinity College, Cambridge
Moshe Vardi	17.08.98–22.08.98	Rice University
Neil Jones	17.08.98–22.08.98	University Kopenhagen
Dov Gabbay	17.08.98–28.08.98	King's College London
Ian Pratt	03.09.98–15.09.98	University of Manchester
Damian Niwinski	09.09.98–22.09.98	University of Warsaw
E. R. Olderog	24.08.98–28.08.98	Universität Oldenburg
Leon van der Torre	10.09.98–12.09.99	IRIT Toulouse
Maurice Margenstern	23.09.98	Université Metz
Yannis Dimopoulos	28.09.98–10 12.98	University of Cyprus
Madala R.K. Krishna Rao	08.10.98	CIT Brisbane
Thomas Hillenbrand	13.10.98	Universität Kaiserslautern
Leon van der Torre	16.10.98–17.10.99	IRIT Toulouse
Jens Knoop	03.11.98–05.11.98	Universität Dortmund
Maurice Nivat	15.11.98–18.11.98	LIAFA Paris 7
Bernhard Steffen	18.11.98–20.11.98	Universität Dortmund
Tiziana Margaria	18.11.98–20.11.98	Universität Dortmund
Peter Revesz	23.11.98–25.11.98	University of Nebraska at Lincoln

1999

Javier Esparza	01.02.99–26.02.99	TU München
Leszeck Pacholsky	03.02.99–12.02.99	University of Wroclaw
Wolfgang Heydrich	10.02.99	Universität Hamburg
Ullrich Hustadt	05.03.99–10.03.99	Manchester Metropolitan University
Rana Barua	08.03.99–12.03.99	Indian Statistical Institute
Bernard Boigelot	22.04.99–23.04.99	Université de Liège

3 First-Order Theorem Proving and Term Rewriting

Work in this area has continued to be both theoretical and experimental. On the methodological level we have given a systematic account of the principal paramodulation-based methods (cf. Section 3.1). We have continued our work on combining algebraic and logic methods and extended our methodological repertoire by considering representation theorems also in this context (cf. Section 3.3).

Decidable fragments of first-order and modal logics have continued to be a major topic of investigation. That part of the work that is more directly related to standard methods in automated theorem proving will be described in the Section 3.2 below, additional results are explained in the Sections 7 and 6.

On the experimental side, substantial effort has been devoted to the further development of the SPASS system (cf. Section 10.1). SPASS has continued to be one of the leading ATP systems world-wide with regard to its performance. Apart from an experimental analysis of the behaviour of different systems on the modal logic K (cf. Section 7), we have also started to use the system in specific applications domains such as the analysis of security protocols (cf. Section 3.4).

3.1 Deduction Systems

Investigators: Leo Bachmair, Harald Ganzinger, Andreas Nonnengart, Andrei Voronkov, Christoph Weidenbach, Jinzhao Wu

Strict Basic Superposition We have solved a long-standing open problem by showing that strict superposition—that is, superposition without equality factoring—is refutationally complete. The calculus was introduced in [17] but its refutational completeness has been an open problem since. The difficulty of the problem arises from the fact that the strict calculus, in contrast to the standard calculus with equality factoring, is not compatible with arbitrary removal of tautologies, so that the usual techniques for proving the (refutational) completeness of paramodulation calculi are not directly applicable. In [6] we have dealt with the problem by introducing a suitable notion of direct rewrite proof and modifying proof techniques based on candidate models and counterexamples in that we define these concepts in terms of direct provability, not semantic truth. We have also introduced a corresponding concept of redundancy with which strict superposition is compatible and that covers most simplification techniques, though not, of course, removal of all tautologies. Reasoning about the strict calculus, as it has turned out, requires techniques known from the more advanced basic variant of superposition [1, 15]. We have also shown that certain superposition inferences from variables are redundant—a result that has turned out to be an indispensable ingredient in the proofs of our results about equality elimination in [7].

For modularizing the completeness proof we have extracted the main ideas behind our proof method into an abstract concept of candidate models, counterexamples and redundancy. This concept turned out to be helpful in the uniform presentation of various paramodulation-based theorem proving methods in our overview paper [5].

Equality Elimination Brand's method [3] is one of the early methods for equality handling in resolution-based theorem proving. Its main idea is that, by flattening terms into terms of depth at most one, the requirement for compatibility of function application with the equality relation is effectively eliminated. We have refined Brand's method for eliminating equality axioms by imposing ordering constraints on auxiliary variables introduced during the transformation process and, in

addition, by avoiding certain transformations of positive equations with a variable on one side [7]. The refinements are both of theoretical and practical interest. For instance, the second refinement is implemented in the Setheo prover [4] and appears to be critical for its performance on equational problems. The correctness of this variant of Brand's method was an open problem that is solved by our more general results in [7]. The experimental results that we have obtained from a prototype implementation of our proposed method have indicated that dramatic improvements of the proof search with tableau methods are possible through the refined transformation. Our completeness proof is also interesting in that we were able to establish a direct connection to basic paramodulation calculi, and thereby shed new light on the connection between different approaches to equational theorem proving.

Boolean Ring-Based Methods By abstracting from certain common properties of various inference systems for first-order logic, we have derived the notion of well-behaved inference rules [12, 11]. Many known inference system including resolution belong to this category. In addition, we have identified two further such well-behaved calculi that were derived from, respectively, the NC-resolution and the pseudo-remainder computation in the Wu-Ritt method. Both of them use single and parallel overlaps between polynomial representations of first-order formulas. We have shown that well-behaved inference rules are complete, and, in addition, compatible with linear and set-of-support restrictions.

In addition to the fact that algebraic techniques may be conveniently applied, another advantage of our class of inference systems is that it depends only weakly upon the underlying logic. Therefore, it is less difficult to extend it to certain non-classical logics. In this regard, we have focused on annotated and many-valued logics. Specifically, we have presented a well-behaved proof system for annotated logics in [8]. In [14] we have described an algorithm to decide the deduction problem in many-valued logics based on the Wu-Ritt method. In [13] we have discussed the extensions of various closed world assumptions to many-valued logics. The methods are effective for any finitely many-valued logic in a uniform way.

Generating Small Clause Normal Forms It is well-known that the quality of the clausal normal form translation has a great impact on proof search. Attempting to generate small sets of clauses is a heuristics which often has a positive effect in this regard. We have investigated formula renaming, improved Skolemization techniques and simplification rules and have experimentally evaluated the impact of these techniques [9]. Running all TPTP [16] examples, it turned out that – except for a few problems – our techniques significantly improve the performance of an automated reasoning system.

Formula renaming is the replacement of subformulae by new predicate symbols. This technique is well-known and preserves more of the structure of the original formula. Aiming at small CNFs we followed the approach of Boy de la Tour [2], where a subformula is only replaced if this eventually leads to a smaller clause set. In the original formulation this test required the computation of exponentially growing functions making it intractable for some problem domains. We improved this test to a combination of some boolean conditions that can be checked in linear time (with respect to the size of the input formula) and do not require any numeric computation at all [10].

Skolemization is the standard technique to eliminate existentially quantified variables by replacing these with suitable applications of Skolem functions. This, ultimately, leads to formulae in which all quantifications are universal. In the literature we find essentially two kinds of Skolemization techniques which differ mainly in the choice of the argument variables for the Skolem functions. In [10] we have proposed two alternative Skolemization techniques which we call "Optimized Skolem-

ization" and "Strong Skolemization", respectively. The effect of these two new techniques compared to the standard ones can best be observed after the whole clause normal form generation has been completed. Optimized Skolemization produces clauses with fewer literals based on the derivation of non-emptiness and totality properties of relations. Strong Skolemization exploits certain semantic independencies between variables so that some arguments to Skolem functions can be replaced by fresh variables. Both of the two new approaches have shown to have a considerable impact on resolution-based theorem provers.

For problems containing equality we have investigated a set of simplification rules that eliminate occurrences of equations [10]. For example, the rule

$$\forall x\,[x \approx t \supset \psi] \rightarrow \psi\{x \mapsto t\}$$

can be used to remove the equation $x \approx t$ if x does not occur in t. For some classes of problems like encodings of planning problems or data type specifications, cardinality properties of minimal models are known or can be easily derived. This can be exploited by further simplification rules. For example, if we know that any minimal model of some formula has at least two domain elements, the rule

$$\forall x\,[x \not\approx t \wedge \phi] \rightarrow \bot$$

falsifies an entire (sub)formula provided x does not occur in t.

References

[1] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995. Revised version of MPI-I-93-236, 1993.

[2] T. Boy de la Tour. An optimality result for clause form translation. *Journal of Symbolic Computation*, 14:283–301, 1992.

[3] D. Brand. Proving theorems with the modification method. *SIAM J. Comput.*, 4:412–430, 1975.

[4] R. Letz, J. Schumann, S. Bayerl, and W. Bibel. Setheo: A high-performance theorem prover. *Journal of Automated Reasoning*, 8(2):183–212, 1992.

- [5] L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction: A Basis for Applications*, volume I, chapter 11, pages 353–397. Kluwer, Dordrecht, Netherlands, 1998.

- [6] L. Bachmair and H. Ganzinger. Strict basic superposition. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 160–174, Lindau, Germany, July 1998. Springer.

- [7] L. Bachmair, H. Ganzinger, and A. Voronkov. Elimination of equality via transformation with ordering constraints. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 175–190, Lindau, Germany, July 1998. Springer. Short version of MPI-I-97-2-012.

- [8] M. Lu and J. Wu. On theorem proving in annotated logics. *Journal of Applied Non-Classical Logics*, 1999. To appear.

- [9] A. Nonnengart, G. Rock, and C. Weidenbach. On generating small clause normal forms. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 397–411, Lindau, Germany, July 1998. Springer.

- [10] A. Nonnengart and C. Weidenbach. On generating small clause normal forms. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, Netherlands, 1999. To appear.

- [11] J. Wu. First-order polynomial based theorem proving. In X.-S. Gao and D. Wang, editors, *Mathematics Mechanizations and Applications*, page 21. Academic, London, 1999.
- [12] J. Wu and Z. Liu. Well-behaved inference rules for first-order theorem proving. *Journal of Automated Reasoning*, 21(3):381–400, 1998.
- [13] J. Wu and M. Lu. CWA in multi-valued logics. In Z. Li, editor, *Proceedings of the 3rd Asian Symposium on Computer Mathematics (ASCM-98)*, pages 259–270, Lanzhou, China, August 1998. Lanzhou University, the Mathematics Mechanization Research Center (MMRC), and the Japanese Society for Symbolic and Algebraic Computation (JSSAC)., Lanzhou University.
- [14] J. Wu, H. Tan, and Y. Li. An algebraic method to decide the deduction problem in many-valued logics. *Journal of Applied Non-Classical Logics*, 8(4):353–360, 1998.

[15] R. Nieuwenhuis and A. Rubio. Theorem proving with ordering and equality constrained clauses. *J. Symbolic Computation*, 19(4):321–352, 1995.

[16] G. Sutcliffe and C. B. Suttner. The tptp problem library – cnf release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

[17] H. Zhang and D. Kapur. First-order theorem proving using conditional rewrite rules. In E. Lusk and R. Overbeek, editors, *Proc. 9th Int. Conf. on Automated Deduction*, volume 310 of *Lecture Notes in Computer Science*, pages 1–20, Berlin, 1988. Springer-Verlag.

3.2 Decision Procedures

Investigators: Harald Ganzinger, Ullrich Hustadt, Florent Jacquemard, Hans de Nivelle, Christoph Meyer, Renate A. Schmidt, Margus Veanes, Christoph Weidenbach

The value of theoretical completeness results about general deduction calculi for first-order logic can partly be measured by the capabilities they provide for obtaining in a uniform way decision procedures for decidable fragments. We have continued to explore the capabilities of standard calculi for first-order logic (superposition, ordered chaining, semantic tableau) into this direction. Our main new results, which are described in more detail below, may be summarized as follows:

(i) We have shown that the [loosely] guarded fragment with equality can be decided by a surprisingly simple instance of superposition, and that the decision procedure is theoretically optimal.

(ii) We have analyzed decidability and complexity issues related to extensions of shallow equational theories.

(iii) For the first time we have given a saturation-based decision procedure for certain (modal) logics with transitive (possibly non-symmetric) relations as an instance of ordered chaining.

(iv) We have shown by exploiting the decidability of (non-simultaneous) rigid E-unification that the $\forall^*\exists\forall^*$ fragment of intuitionistic logic with equality is decidable by tableau methods.

The advantage of inference-based decision procedures over semantic procedures based on collapsing models is that the former use syntactic, unification-based inferences to enumerate candidate witnesses of inconsistency. There is experimental evidence [16] that such inference-based procedures perform well in practice, in particular they often will not exhibit the usually exponential or double-exponential worst-case complexity of the respective fragments. Also, when having a flexible saturation theorem prover such as SPASS (cf. section 10.1) at hand it suffices to appropriately adjust its parameters in order to efficiently implement the procedure.

The Guarded Fragment The guarded fragment was introduced in [1] as 'the modal fragment of classical logic'. It is a function-free fragment in which quantification is relativized in the form

$$\forall y[R(x,y) \supset A(x,y)] \text{ and } \exists y[R(x,y) \wedge A(x,y)].$$

The atoms $R(x, y)$ which, in the general case, have to contain all the free variables of the body of the quantification, are called guards. These forms naturally arise when modal formulae are translated into classical logic using the standard translation based on the Kripke frames. The guarded fragment retains many of the nice properties of modal logics, including the tree model property and decidability. Any decision procedure for this fragment, hence, is a decision procedure for those modal logics that can be embedded into it, in particular $K, D, S3$, and B. In [7] it was shown that equality can be admitted in the guarded fragment without affecting decidability. In the fragment with equality additional logics such as difference logic can be expressed (where $\Diamond A$ means A holds in a world different from the present).

In [5] a resolution decision procedure was given for the guarded fragment without equality. In this procedure, a non-liftable ordering is employed, and, hence, some additional and non-trivial argument was required for proving refutational completeness. In [11] we have now presented a decision procedure for the guarded fragment with equality which is based on resolution and superposition. Despite the fact that it applies to a larger fragment, our new procedure is simpler than the one in [5] in that we employ a liftable ordering (plus selection) so that we are able to re-use standard results about refutational completeness. Our method is also interesting as there are not so many saturation-based decision procedures for fragments with equality described in the literature. Furthermore, the worst-case time complexity of the decision procedure is double exponential which is optimal, given that the logic is 2EXPTIME-complete [7]. We were able to extend the method to the loosely guarded fragment with equality, where it becomes technically much more involved. In the loosely guarded case, a conjunction of atoms may serve as a guard, provided certain co-occurrence requirements for free variables are satisfied. For the extension, hyper-inferences which simultaneously resolve a conjunction of guard atoms were needed. Some non-trivial results were required about the existence of suitable partial inferences to avoid the generation of clauses which are too deep, together with meta-theorems about the refutational completeness of these partial inferences.

Extensions of Shallow Equational Theories Semantic methods for proof search are based on the truth value of clauses with respect to certain interpretations, called model hypotheses. In [14] we have proposed a method called soft typing which combines this idea into resolution and superposition theorem proving. The model hypotheses there are exactly the ones on which the theoretical completeness proofs of these calculi are based upon. In its full generality the approach is merely a theoretical concept. The interpretations might not be effectively representable or the truth value of a universally quantified clause might not be decidable. Therefore, in general one employs suitable approximations. Sorted equational theories appear to be useful candidates in this regard. This motivates our interest in decidability questions related to sorted equational theories. A detailed exposition of the soft typing techniques as well as a comprehensive discussion of sorted equational theories is currently in preparation [8].

Our main result [19] in this area is the decidability of unification with respect to so-called semi-linear sorted equational theories. Basically, a semi-linear theory is given by identities in which non-linear variables only appear in the same subterms. For example, the equation $f(f(x, x), y) \approx g(f(x, x))$ is semi-linear whereas $f(g(x), h(x)) \approx h(g(x))$ is not. The result was obtained by first transforming a semi-linear theory into an essentially equivalent so-called shallow sorted equational theory in which all equations are shallow. An equation is shallow if any proper subterm in the equation is a variable, e.g. the equation $f(x, x) \approx g(x)$ is shallow. The transformation was originally suggested by Uribe [25] in the context of set constraints and exploited by Weidenbach [22] for sort theories. We have also proved that shallow sorted equational theories can be finitely saturated by a

certain refinement of superposition with selection. Unifiability with respect to a saturated shallow sorted equational theory is then shown to be decidable.

Shallow sorted equational theories naturally generalize tree automata with equality constraints [2] as well as shallow sort theories [22]. The saturation of shallow sorted equational theories under superposition is related to techniques based on the completion of (standard) tree automata as proposed by Comon [3]. However, we have demonstrated that tree automata techniques, even with extended concepts such as tree automata with equality constraints [2], are not sufficient for our purpose.

Semi-linear sorted equational theories strictly include the shallow theories as suggested in [4]. They are related to Nieuwenhuis' standard theories [23], but do not strictly generalize these. In [19] we also discuss a generalization of semi-linear sorted equational theories which strictly embeds the standard theories. There pseudo-linear extensions, however, have turned out to be undecidable.

Transitive Propositional Modal Logics Transitive propositional modal logics are characterized by the iterated modality in the schema $\mathsf{4} = \Box p \supset \Box\Box p$. In contrast to systems like K, KD or KT, the number of modal operators does not diminish during deduction in Hilbert calculi due to the schema 4. In order to avoid unlimited derivations, some form of cycle detection mechanism is therefore essential when using tableaux-like calculi or modal resolution calculi. In the semantics-based translation approaches, modal formulae are embedded into first-order logic and conventional first-order theorem proving is applied henceafter. Here the difficulty is caused by transitivity clause $\forall x, y, z\ (R(x, y) \wedge R(y, z) \supset R(x, z))$ which leads, in general, to unlimited growth of the size of formulae. It is possible to use pre-computed term depth bounds, whereby termination can be guaranteed [21, 20]. However, in practice this approach performs poorly [18]. For non-transitive modal logics, good performance results have been obtained with the resolution theorem prover SPASS [16, 17, 18, 20] (cf. section 10.1). A general term rewriting-based calculus designed for binary relations satisfying the composition laws of the form $R \circ R' \subseteq R''$ (including equality) was given by Bachmair and Ganzinger [9], and combines ideas from term rewriting and resolution in a calculus of ordered chaining. A specific problem with non-symmetric transitive relations is that chaining into variables cannot be avoided in general. Fortunately, in ordered chaining these problems only arise with unshielded variables. Yet, there was no natural fragment known for which ordered chaining yields a decision procedure.

In [12] we have now shown how ordered chaining may be used to obtain a saturation decision procedures for the relational translation of a range of transitive propositional modal logics, in particular, of the logics K4, KD4, and S4. The method may be applied also to multi-modal logics with modal operators satisfying (a subset of) D, T, and 4 as well as combinations thereof. The important ingredients of our method are structural CNF transformation and ordered chaining with selection. Structural transformation allows us to embed the logics and formulae under consideration into a well-behaved class of clauses. Mechanisms like cycle detection or enumerating all clauses up to a pre-computed size bound are not required. Our solution requires no specialized techniques, only standard theorem proving techniques are used. The whole effort has been to find a suitable ordering and selection function so as to ensure termination for extensions of K4.

Encouraged by this result we have tried to extend the method to the monadic guarded fragment with equality and transitive relations. In this variant of the guarded fragment (GF), binary relations may be specified as transitive, but non-monadic predicates, and in particular the transitive relations, are only allowed to occur in guards. In the monadic GF one may have clauses of the form $\forall x\,(C \vee R(x, f(x)))$ as well as $\forall x\,(C \vee R(f(x), x))$ whereas in the modal fragment formulas of the latter kind do not occur. Although the monadic GF with transitive relations was found to be decidable by

other methods [13], cf. Section 6, we were not able to deal with the resulting technical complications in the chaining framework.

Intuitionistic Predicate Logic with Equality Herbrand's theorem plays a fundamental role in automated theorem proving methods based on tableaux. The crucial step in procedures based on such methods can be described as the corroboration problem or the Herbrand skeleton problem, where, given a positive integer m, called multiplicity, and a quantifier free formula, one seeks a valid disjunction of m instantiations of that formula. In the presence of equality, which is the case here, corroboration with multiplicity 1 is closely related to a problem called simultaneous rigid E-unification (SREU) that was initially proposed to handle equality in tableaux or matrix methods [6].

In [15] the corroboration problem and SREU are studied in detail, and new elementary undecidability proofs are presented for new restricted cases of those problems. The main contributions are two theorems. The first, the Partisan Corroboration Theorem, relates corroboration problems with different multiplicities. The second, the Shifted Pairing Theorem, is a finite tree automata formalization of a technique that was introduced in [24] for proving undecidability results through direct encodings of Turing machine computations. These theorems are used in [15] to explain and sharpen several recent results related to the corroboration problem, the simultaneous rigid E-unification problem and the prenex fragment of intuitionistic logic with equality.

In [10] SREU is shown to be decidable, and in fact EXPTIME-complete, if only a single variable is allowed. This result is used to prove that the $\forall^*\exists\forall^*$ fragment of intuitionistic logic with equality is decidable. This is in contrast with the undecidability of SREU with two variables, and the undecidability of the $\exists\exists$-fragment [26]. Altogether one now has a complete classification of decidability for prenex fragments of intuitionistic logic with equality, in terms of the quantifier prefix.

References

[1] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. Research Report ML-96-03, ILLC, 1996.

[2] B. Bogaert and S. Tison. Equality and disequality constraints on direct subterms in tree automata. In A. Finkel and M. Jantzen, editors, *Proceedings of Symposion on Theoretical Aspects of Computer Science (STACS '92)*, volume 577 of *LNCS*, pages 161–172. Springer, February 1992.

[3] H. Comon. Sequentiality, second order monadic logic and tree automata. In *Proceedings 10th IEEE Symposium on Logic in Computer Science, LICS'95*, pages 508–517. IEEE Computer Society Press, 1995.

[4] H. Comon, M. Haberstrau, and J. P. Jouannaud. Syntacticness, cycle-syntacticness and shallow theories. *Information and Computation*, 111(1):154 – 191, 1994.

[5] H. de Nivelle. A resolution decision procedure for the guarded fragment. In C. Kirchner and H. Kirchner, editors, *CADE-15*, lncs, pages 191–204. Springer-Verlag, 1998.

[6] J. Gallier, S. Raatz, and W. Snyder. Theorem proving using rigid E-unification: Equational matings. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, 1987.

[7] E. Grädel. On the restraining power of guards. Manuscript, 1997.

[8] C. Meyer. *Soft Typing for Clausal Inference Systems*. Dissertation, Technische Fakultät der Universität des Saarlandes, Saarbrücken, Germany, 1999. in preparation.

- [9] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998. Revised Version of MPI-I-95-2-009.

- [10] A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid e-unification with one variable. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 181–195, Tsukuba, Japan, March 1998. Springer.
- [11] H. Ganzinger and H. De Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999. To appear.
- [12] H. Ganzinger, U. Hustadt, C. Meyer, and R. A. Schmidt. A resolution-based decision procedure for extensions of K4. In *Advances in Modal Logic, Volume 2*, CSLI Lecture Notes. CSLI, Stanford, USA, June 1999. To appear.
- [13] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999. To appear.
- [14] H. Ganzinger, C. Meyer, and C. Weidenbach. Soft typing for ordered resolution. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (CADE-14)*, volume 1249 of *Lecture Notes in Computer Science*, pages 321–335, Townsville, Australia, 1997. Springer.
- [15] Y. Gurevich and M. Veanes. Logic with equality: Partisan corroboration, and shifted pairing. *Information and Computation*, 1999. To appear.
- [16] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logics. In M. E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 202–207, Nagoya, Japan, 1997. International Joint Conferences on Artificial Intelligence, Inc. (IJCAII) and Japanese Society for Artificial Intelligence (JSAI), Morgan Kaufmann.
- [17] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, page 42, 1999. To appear.
- [18] U. Hustadt, R. A. Schmidt, and C. Weidenbach. Optimised functional translation and resolution. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reaso ning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 36–37, Oisterwijk, The Netherlands, May 1998. Springer.
- [19] F. Jacquemard, C. Meyer, and C. Weidenbach. Unification in extensions of shallow equational theories. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 76–90, Tsukuba, Japan, 1998. Springer.
- [20] R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, volume 87 of *CSLI Lecture Notes*, chapter 13, pages 189–208. CSLI, Stanford, USA, 1998.
- [21] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.
- [22] C. Weidenbach. Sorted unification and tree automata. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction - A Basis for Applications*, Applied Logic, chapter 9, pages 291–320. Kluwer, Dordrecht, Netherlands, 1998.

[23] R. Nieuwenhuis. Basic paramodulation and decidable theories (extended abstract). In *Proceedings 11th IEEE Symposium on Logic in Computer Science, LICS'96*, pages 473–482. IEEE Computer Society Press, 1996.

[24] D. Plaisted. Special cases and substitutes for rigid E-unification. Technical Report MPI-I-95-2-010, Max-Planck-Institut für Informatik, November 1995.

[25] T. E. Uribe. Sorted unification using set constraints. In *11th International Conference on Automated Deduction, CADE-11*, volume 607 of *LNCS*, pages 163–177. Springer, 1992.

[26] M. Veanes. The undecidability of simultaneous rigid E-unification with two variables. In *Proc. Kurt Gödel Colloquium KGC'97*, volume 1289 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 1997.

3.3 Combination of Algebraic and Logic Methods

Investigators: Viorica Sofronie-Stokkermans, Georg Struth, Jürgen Stuber, Uwe Waldmann

We have continued to work on combining algebraic and logic methods in automated theorem proving for varieties of groups, rings, fields, and lattices. In the previous period, most of our approaches were centered around specializations of superposition and chaining. Equational reasoning in the presence of the associativity and commutativity axioms is known to be difficult – theoretically as well as in practice. Using AC-unification and extended clauses the worst inefficiencies of a naïve approach can be avoided, but still the need for extended clauses leads to numerous variable overlaps, one of the most prolific types of inferences in resolution or superposition style calculi. The hope was that more specialized calculi would avoid many, if not all, of these implicit variable overlaps. Inequality reasoning as it is required for reasoning about orderings, in particular lattices, suffers from related problems. There, even overlaps below variables have to be considered.

Within the period reported here, our methodological repertoire with regard to dealing with lattices was extended by model-theoretic concepts based on representation theorems. The main idea behind representation theorems is to decompose a given algebraic structure into simpler structures. Usually, this makes it possible to better understand the properties of these structure.

The Priestley representation theorem, for instance, states that every bounded distributive lattice A is isomorphic to the lattice of all continuous, order-preserving maps from its Priestley dual $D(A)$ (a partially-ordered topological space) into the partially-ordered set with two elements. Thus, every bounded distributive lattice A can be represented as a sublattice of a direct product of 2-element lattices, where the index set is partially-ordered. The main goal of our research in this area was to study and extend the Priestley representation theorem for bounded distributive lattices, and to investigate its applications, in particular by developing automated theorem proving procedures for certain classes of non-classical logics, or, more generally, for varieties of distributive lattices with operators.

Rewrite-Based Techniques for Abelian Groups, Rings, and Lattices The problems caused by the AC axioms can be mitigated by integrating more algebraic structure: In the presence of the axioms of Abelian groups or at least cancellative Abelian monoids, ordering restrictions render most variable overlaps unnecessary, and it remains to consider inferences with unshielded variables, that is, variables not occurring anywhere below a free function symbol (Waldmann [27]).

Divisible torsion-free abelian groups (e. g., the rational or real numbers) allow quantifier elimination: For every quantified formula over 0, +, and $\approx$ there exists a quantifier-free formula that is equivalent modulo the theory axioms. In particular, every closed formula over this vocabulary is provably true or false: the theory of divisible torsion-free abelian groups is complete and decidable. Superposition calculi, however, work on formulae that do not contain any existential quantifiers, but that may contain free function symbols – possibly introduced by Skolemization, possibly given initially. In the presence of free function symbols, and possibly other sorts, there is of course no way to eliminate all variables from a formula – not even all universally quantified ones – but we can at least give an effective method to eliminate all unshielded variables.

This elimination algorithm has been integrated into the cancellative superposition calculus in (Waldmann [24, 25, 27]). The resulting calculus is a decision procedure if there is only one sort

and all free functions are the result of Skolemization (Waldmann [28]). In the presence of arbitrary free function symbols and additional sorts it is still refutationally complete [27]. The integration of the elimination algorithm allows us to dispense with variable overlaps altogether. Using eager abstraction, it is also possible to avoid the computation of ACU unifiers and ACU orderings. The latter result is a consequence of the fact that every reduction ordering over terms not containing + that is total on ground terms and for which 0 is minimal can be extended to an ordering that is ACU-compatible and has the multiset property (Waldmann [26]).

In the context of commutative rings and algebras technical problems arise due to AC-extensions for multiplication. In the partial interpretations used in our completeness proof, transitivity, or equivalently the Church-Rosser property, holds only for terms below a certain bound with respect to the given term ordering. It is therefore necessary to construct equational proofs that stay below this bound. By combining the equivalence of normalized and unrestricted rewriting for systems that are Church-Rosser with the notions of strong symmetrization and semi-compatibility we found a technique that is powerful enough to carry out our proofs [20]. This is for instance used in a detailed exposition of the case of commutative rings [22].

We have also considered the case of modules over integers [21]. In the cases of modules and of algebras we need special orderings that cannot be obtained by combining orderings previously found in the literature. To simplify the construction of suitable orderings we have developed the notion of a theory path ordering [23] that is a generalization of the associative path ordering. To define an ordering it suffices to give an ordering over interpreted function symbols and constants, which is then extended to the whole signature by a precedence on free function symbols, which are assumed to have lexicographic status.

In [19] the combinatorial theory of non-symmetric rewriting (modulo congruences) is applied to obtaining new rewriting-based proof calculi for semilattices and distributive lattices. In particular, ordered resolution can be reconstructed as ordered chaining (modulo AC) for distributive and Boolean lattices.

Representation Theorems for Distributive Lattices with Operators The research in this direction has as its goal the better understanding of the link between the algebraic and the Kripke-style models for certain classes of non-classical logics. In [17] we made a first step in this direction, by developing a Priestley duality theorem for a class of algebras called SHn-algebras (Symmetric Heyting algebras of order n) introduced and studied by Luisa Iturrioz [6, 7], who used them in the investigation of SHn-logics, an extension of Łukasiewicz logics [6] (a brief presentation of the properties of SHn-algebras and SHn-logics can be also found e.g. in [9]). We have shown that this Priestley-style duality helps in proving in a direct way the soundness and completeness of SHn-logics with respect to a class of Kripke-style models similar to those introduced by Iturrioz and Orłowska in [8], by using only soundness and completeness of SHn-logics with respect to the variety of SHn-algebras. Subsequently we have extended these results to more general classes of algebras. In [14] we have given a Priestley-type duality for distributive lattices endowed with a general class of well-behaved operators, including various types of anti(hemi)morphisms. This extends the Priestley duality theorem for bounded distributive lattices [29, 30] as well as the Priestley-style duality theorem for distributive lattices endowed with hemimorphisms established by Goldblatt in [3]. We also have shown that finitely-generated varieties of distributive lattices with operators are closed under canonical embedding algebras. The results have been used in [15] to construct topological and non-topological Kripke-style models for logics that are sound and complete with respect to varieties of distributive lattices with operators in the above-mentioned classes.

Theorem Proving in Varieties of Distributive Lattices with Operators In [16] we have proposed a method for automated theorem proving in the universal theory of certain varieties of distributive lattices with well-behaved operators. We have exploited our extension of Priestley's representation theorem for distributive lattices in [14] to establish a link between satisfiability of universal sentences with respect to varieties of distributive lattices with operators, and satisfiability with respect to certain classes of relational structures. This has resulted in a new embedding for universal sentences over such varieties into clauses over certain quasi-orderings. As a result, some of the more critical aspects of the lattice structure are lifted to related structures on the level of first-order clause logic where they can be dealt with by standard methods. In particular, saturation-based techniques for theories of reflexive and transitive relations, such as ordered chaining with selection [10], can then be applied successfully. Decidability and complexity results follow in many cases as consequences of existing decision procedures based on ordered resolution or ordered chaining. The embedding is structure-preserving because one can establish direct correspondences between the structure of the sets of clauses that are generated with our method, and certain algebraic properties of the original varieties.

The embedding into classical logic can, in particular, be used for automated theorem proving in many classes of non-classical propositional logics. In [18] we have analyzed a certain class of logics which can be proved sound and complete with respect to classes of distributive lattices with operators. We have shown that, given such a logic $\mathcal{L}$, very often properties of their Priestley duals can be exploited to define a class of Kripke-style models with respect to which $\mathcal{L}$ is sound and complete. If this class of Kripke-style models is elementary, it may then form the basis for applying standard first-order methods such as resolution. Preliminary results (where, however, the emphasis was on finitely-valued logics) were also presented in [12].

In finitely-valued logics the situation is even simpler. If $\mathcal{L}$ is a finitely-valued logic having as algebra of truth values a finite distributive lattice with operators A, then the Priestley dual $D(A)$ of A can be seen as a finite Kripke-style frame with respect to which $\mathcal{L}$ is sound and complete. Since A is finite, $D(A)$ is in bijective correspondence with the set of join-irreducible elements of A, hence has fewer elements than A itself. Exploiting this fact in the CNF transformation, in many cases fewer clauses are generated than with very general procedures, such as those described in [1] or with procedures which are based on sets-as-signs (cf., e.g., [4, 2]). Moreover, our results show that only principal filters generated by join-irreducible elements of A and their complements are needed as signs for that latter approach. Even a version of first-order finitely-valued logics with quantifiers $\forall, \exists$ (interpreted as generalized meets resp. joins in A), defined by following the ideas in [1], is tractable this way. These ideas have been illustrated by means of an example in [13], and presented in their full generality in [32], where we have extend our earlier results in [31] considerably. Influenced by ongoing research in many-valued logics, in these papers we regarded the resulting sets of clauses as "signed clauses". In [11] we have shown that the translation to clause form is actually a translation to classical logic, and that soundness and completeness of various refinements of the (signed) resolution procedure, as well as decidability and complexity results for certain classes of signed clauses, follow as a consequence of results from first-order logic. This explains and extends earlier results on theorem proving in finitely-valued logics, and especially regular logics [5].

Thus, the method in [16] subsumes the standard methods for embedding modal logics in classical logic, as well as our extension to more general non-classical logics described in [18], and also some of our methods for automated theorem proving in finitely-valued logics based on distributive lattices with operators [13, 32].

References

[1] M. Baaz and C. Fermüller. Resolution-based theorem proving for many-valued logics. *Journal of Symbolic Computation*, 19:353–391, 1995.

[2] M. Baaz, C. Fermüller, and G. Salzer. Automated deduction for many-valued logics. In A. Robinson and A. Voronkow, editors, *Handbook of Automated Reasoning*. Elsevier Science and MIT Press, to appear 1999.

[3] R. Goldblatt. Varieties of complex algebras. *Annals of Pure and Applied Logic*, 44(3):153–301, 1989.

[4] R. Hähnle. Short conjunctive normal forms in finitely valued logics. *Journal of Logic and Computation*, 4(6):905–927, 1994.

[5] R. Hähnle. Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6(1):49–69, 1996.

[6] L. Iturrioz. Modal operators on symmetrical Heyting algebras. In T. Traczyk, editor, *Universal Algebra and Applications*, Banach Center Publications, Vol.9, pages 289–303. PWN-Polish Scientific Publishers, 1982.

[7] L. Iturrioz. Symmetrical Heyting algebras with operators. *Zeitschrift f. math. Logik und Grundlagen d. Mathematik*, 29:33–70, 1983.

[8] L. Iturrioz and E. Orłowska. A Kripke-style and relational semantics for logics based on Łukasiewicz algebras. Conference in honour of J. Łukasiewicz, Dublin, 1996.

[9] L. Iturrioz and V. Sofronie-Stokkermans. SHn-algebras. Atlas of Many-Valued Logics, 1999. Submitted.

- [10] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998. Revised Version of MPI-I-95-2-009.
- [11] V. Sofronie-Stokkermans. On translation of finitely-valued logics to classical first-order logic. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 410–411, Brighton, UK, 1998. Wiley.
- [12] V. Sofronie-Stokkermans. Representation theorems and automated theorem proving in certain classes of non-classical logics. In P. Eklund, G. Escalada-Imaz, R. Haehnle, and P. Vojtas, editors, *Proceedings of the Workshop on Many-Valued Logic for AI Applications (ECAI-98)*, Brighton, UK, August 1998. ECAI.
- [13] V. Sofronie-Stokkermans. Resolution-based theorem proving for shn-logics. Technical Report E1852-GS-981, Technische Universität Wien, Vienna, Austria, November 1998. an extended version will appear in LNCS (subseries LNAI); Proceedings of FTP'98.
- [14] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators and applications to the semantics of non-classical logics. part i. *Studia Logica*, 1999. To appear.
- [15] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators and applications to the semantics of non-classical logics. part ii. *Studia Logica*, 1999. To appear.
- [16] V. Sofronie-Stokkermans. On the universal theory of varieties of distributive lattices with operators: Some decidability and complexity results. In *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, Lecture Notes in Computer Science, Trento, Italy, 1999. Springer. To appear.
- [17] V. Sofronie-Stokkermans. Priestley duality for SHn-algebras and applications to the study of Kripke-style models for SHn-logics. *Multiple-Valued Logic - An International Journal*, 1999. To appear.

- [18] V. Sofronie-Stokkermans. Representation theorems and automated theorem proving in non-classical logics. In *Proceedings of the 29th IEEE International Symposium on Multiple-Valued Logic (ISMVL 99)*, Freiburg im Breisgau, Germany, 1999. IEEE Computer Society Press. To appear.

- [19] G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic.* PhD thesis, Universität des Saarlandes, April 1999.

- [20] J. Stuber. Strong symmetrization, semi-compatibility of normalized rewriting and first-order theorem proving. In M. P. Bonacina and U. Furbach, editors, *Proceedings of the International Workshop on First-Order Theorem Proving (FTP-97)*, volume 97-50 of *RISC-Linz Report*, pages 125–129, Schloss Hagenberg by Linz, Austria, 1997. Johannes Kepler Universität.

- [21] J. Stuber. Superposition theorem proving for abelian groups represented as integer modules. *Theoretical Computer Science*, 208(1-2):149–177, 1998.

- [22] J. Stuber. Superposition theorem proving for commutative rings. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction - A Basis for Applications. Volume III. Applications*, volume 10 of *Applied Logic Series*, chapter 2, pages 31–55. Kluwer, Dordrecht, Netherlands, 1998.

- [23] J. Stuber. Theory path orderings. In *Tenth International Conference on Rewriting Techniques and Applications (RTA99)*, Lecture Notes in Computer Science, Trento, Italy, 1999. Springer. To appear.

- [24] U. Waldmann. *Cancellative Abelian Monoids in Refutational Theorem Proving.* PhD thesis, Universität des Saarlandes, July 1997.

- [25] U. Waldmann. A superposition calculus for divisible torsion-free abelian groups. In M. P. Bonacina and U. Furbach, editors, *Proceedings of the International Workshop on First-Order Theorem Proving (FTP-97)*, number 97-50 in RISC-Linz Report Series, pages 130–134, Linz, Austria, 1997. Johannes Kepler Universität.

- [26] U. Waldmann. Extending reduction orderings to ACU-compatible reduction orderings. *Information Processing Letters*, 67(1):43–49, 1998.

- [27] U. Waldmann. Superposition for divisible torsion-free abelian groups. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 144–159, Lindau, Germany, 1998. Springer.

- [28] U. Waldmann. Cancellative superposition decides the theory of divisible torsion-free abelian groups. Research Report MPI-I-1999-2-003, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, March 1999.

[29] H. Priestley. Representation of distributive lattices by means of ordered Stone spaces. *Bull. London Math Soc.*, 2:186–190, 1970.

[30] H. Priestley. Ordered topological spaces and the representation of distributive lattices. *Proc. London Math. Soc.*, 3:507–530, 1972.

[31] V. Sofronie-Stokkermans. *Fibered Structures and Applications to Automated Theorem Proving in Certain Classes of Finitely-Valued Logics and to Modeling Interacting Systems.* PhD thesis, RISC-Linz, J.Kepler University Linz, Austria, 1997.

[32] V. Sofronie-Stokkermans. Automated theorem proving by resolution for finitely-valued logics based on distributive lattices with operators. *Multiple-Valued Logic - An International Journal*, 1999. To appear.

3.4 Analyzing Security Protocols in First-Order Logic

Investigator: Christoph Weidenbach

The growing importance of the internet causes a growing need for security protocols that protect transactions and communication. It turns out that the design of such protocols is highly error-prone. Therefore, a variety of different methods have been described that analyze security protocols to discover flaws. Well known methods are based on model checking [1] or induction [4]. The former eventually maps the potentially infinite state (message) space to a finite state space where properties can then be shown automatically, whereas the latter uses explicit induction (supported by the higher-order theorem prover Isabelle) on the infinite state (message) space as its reasoning paradigm. The inductive method allows for subtle protocol models, but requires user interaction at proof time. Our approach tries to combine the benefits of both techniques. We use (decidable) monadic fragments of first-order logic as a specification language. The language is sufficiently rich to model infinite state (message) spaces via its usual minimal model(s) semantics, but reasoning remains automatic via automated theorem proving. Following this approach we have successfully analyzed the Neuman-Stubblebine key exchange protocol [3] and, in cooperation with the cryptography group at the university Saarbrücken, a recent signature signing protocol [5] using our automated theorem prover SPASS [2] (cf. section 10.1).

To illustrate our method, a typical formula describing an intruder which is able to records all messages that are sent is

$$\forall x, y, z[M(sent(x, y, z)) \supset Im(z)]$$

where the term $sent(x, y, z)$ represents the message z sent from participant x to participant y. M holds precisely for all sent messages, and Im represents the set of messages the intruder is able to record (and, therefore, to resend, or to decompose into their components). Formulas of a similar structure can be employed to describe that an intruder is able to compose/decompose or encrypt/decrypt messages. We were able to prove that the first-order theory with respect to the minimal model of such an intruder theory is always decidable [2].

References

[1] J. C. Mitchell. Finite-state analysis of security protocols. In A. J. Hu and M. Y. Vardi, editors, *Computer Aided Verification (CAV-98) : 10th International Conference*, volume 1427 of *LNCS*, pages 71–76. Springer, 1998.

- [2] C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 314–328, Trento, Italy, 1999. Springer. To appear.

[3] B. C. Neuman and S. G. Stubblebine. A note on the use of timestamps as nonces. *ACM SIGOPS, Operating Systems Review*, 27(2):10–14, 1993.

[4] L. C. Paulson. Proving properties of security protocols by induction. In J. Millen, editor, *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 70–83. IEEE Computer Society, 1997.

[5] B. Pfitzmann, M. Schunter, and M. Waidner. Efficiency of optimistic contract signing. In *17th Symposium on Principles of Distributed Computing, PODC-98*, pages 113–122. ACM Press, 1998.

4 Logic Methods for Program Analysis

The goal of program analysis is to statically compute (possibly partial) answers to questions about the runtime behavior of a program. Such questions may relate to data structures at specific program points as in data flow analysis ("Does an expression always have a constant value?") or to the overall runtime behavior of program executions (viz. a temporal property, such as termination) as in model checking. We investigate whether logic methods can be useful for this goal. By 'logic methods' we mean techniques, heuristics and (possibly partial) algorithms for solving logical decision problems (as opposed to e.g. graph problems).

Generally, we aim at finding strategies to help avoid redundant computations for the 'proofs' needed in program analysis. We need to find out whether strategies can play a similarly important role for solving difficult problems as they do e.g. in first-order theorem proving. Our efforts here are complementary to the already heavily invested efforts in optimizing data structures (e.g. BDD's).

The motivation to look at logic methods is to be able to formulate algorithms on a high, formally well-founded level. The formulation of an algorithm in terms of logical deduction helps us to devise provably correct optimizations (based on strategies and abstraction). The gap between the data structures used in the problem formulation and its solution, respectively, disappears. Logical formulas yield an appropriate representation of data especially over an infinite domain.

Our work follows two branches. In the one branch, we investigate the fundamental structure of the logical decision problems involved; this branch of work is mostly related to set constraints; see Section 6 for a variety of results.

In the other branch, we investigate how program analysis problems can be translated into logical decision problems in a fruitful way. The translation is fruitful if one gains new insight into the problem by viewing it from a different perspective, or if one can exploit existing optimizations, or if one can reuse existing implementations. All three criteria are fulfilled by our work reported in the following two sections. There, we express the program property to be analyzed as a specific solution of appropriate constraints. We thus reduce the general program analysis problem to the problem of solving a constraint. 'Solving a constraint' is the logical deduction problem of inferring an equivalent constraint in solved form. A 'solved form' allows one to answer the relevant questions about solutions effectively, usually in linear time (that questions are typically reduced to emptiness or membership problems).

Our work opens a new connection between theorem proving, program analysis and model checking. The three techniques: deduction for proofs of validity, fixpoint iteration for the computation of abstract program semantics, and proofs of the satisfaction of temporal formulas on models of system behavior, can be formally related with each other.

We separate the description of our work on program analysis according to the two basic cases of symbolic resp. numeric data structures on which the program behavior may depend. Although the constraint-solving view described above applies to both cases on an abstract level, the respective methods have a quite different flavor.

4.1 Set-based analysis

Investigators: Witold Charatonik, Andreas Podelski

In this section, we report on our work on the first kind of program analysis, where we analyze program behavior that depends on 'symbolic' data structures such as lists, stacks, queues etc.. We use strings or trees to model the data structures. We have shown in [6] that one can translate the analysis problem for such programs to the problem of solving constraints overs sets of tuples of

trees, a problem which we immediately weaken to a decidable problem for so-called set constraints (the corresponding abstraction consists of ignoring tuple dependencies). The solved form of a set constraint is an inductive definition (or a more general fixpoint definition) that is paramount to a notion of automaton (on finite or infinite strings or trees). The corresponding form of program analysis is called set-based.

More specifically, in [6], we show that the set of all system states satisfying a given temporal property specified in Clarke and Emerson's Computation Tree Logic (CTL) can be expressed in terms of least and greatest models of logic programs (which coincides with the denotational program semantics). The idea is that we model tree-valued states as ground atoms and transition systems as logic programs. Here, logic programs are nothing but a syntactic variant of set constraints: the (wlog. unary) predicates are variables over sets of trees. In [6], we show how this translation can be used in two ways. (1) For the general class of systems over tree-like data structures (specified e.g. by while programs), the translation can be used for a type analysis (an abstraction yields a logic program, viz. a set constraint, for which existing constraint solving algorithms [3, 4] yield a type for each program expression; the type approximates the set of values for which a state satisfies a given temporal property). (2) For the special class of pushdown systems that model (imperative) programs with recursive procedure calls, the translation already yields a decidable class of set constraints; i.e., the corresponding algorithms yield a full test of CTL properties.

The idea of analyzing programs with respect to the least and greatest models is extended in [2] to arbitrary fixpoints (in the sense of the full μ-calculus). We introduce the Horn μ-calculus — a logic programming language allowing arbitrary nesting of least and greatest fixed points. We show that in spite of its extreme expressive power, nontrivial static analysis is possible. In particular, we show that a variety of set-based type inference algorithms for logic programs generalize to the Horn μ-programs.

Our original motivation to work on set-based analysis (as explained in the Third Biennial Report of MPI-I) is the error diagnosis for concurrent constraint programming languages, in particular the language Oz being developed in Smolka's group at DFKI and the University of Saarbrücken. In an abstraction step that is too rough for verification but may still be useful for type analysis, one associates a concurrent constraint program with a logic program; the program states are now non-ground atoms (viz. constraints), in contrast with the situation described above, which makes the analysis a more ambitious problem. In [7], we present the first approximation method of the finite-failure set of a logic program by set-based analysis (previous analyses had all considered the success set). We exhibit the connection between finite failure and the inevitability of the 'inconsistent-store' error in fair executions of concurrent constraint programs where no process suspends forever. This way, we obtain a first automated error diagnosis method; its integration into the Oz compiler depends on the practical efficiency of the corresponding constraint solver (the theoretical complexity of the problem is DEXPTIME [4]). We have developed a BDD-based algorithm and are currently working on its implementation.

In [5] we apply methods of set-based analysis to checking or inferring specific types for logic programs that take into account the directionality of predicate uses. We characterize so-called directional types for logic programs in model-theoretic terms. As a consequence, we obtained the first method for inferring directional types. We also improve on previous work by Aiken who gave an NEXPTIME-algorithm for checking a subclass of regular directional types. We give a DEXPTIME algorithm and show that it is optimal. In [1], we extend this result to the full class of regular directional types.

References

[1] W. Charatonik. Directional type checking: Beyond discriminative types. Submitted, May 1999.

- [2] W. Charatonik, D. McAllester, D. Niwinski, A. Podelski, and I. Walukiewicz. The horn mu-calculus. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 58–69, Indianapolis, Indiana, June 1998. IEEE Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.
- [3] W. Charatonik and A. Podelski. Set constraints with intersection. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97)*, pages 362–372, Warsaw, Poland, 1997. IEEE Comput. Soc. Press.
- [4] W. Charatonik and A. Podelski. Co-definite set constraints. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 211–225, Tsukuba, Japan, March 1998. Springer.
- [5] W. Charatonik and A. Podelski. Directional type inference for logic programs. In G. Levi, editor, *Proceedings of the 5th International Symposium in Static Analysis (SAS-98)*, volume 1503 of *Lecture Notes in Computer Science*, pages 278–294, Pisa, Italy, September 1998. Springer.
- [6] W. Charatonik and A. Podelski. Set-based analysis of reactive infinite-state systems. In B. Steffen, editor, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS-98)*, volume 1384 of *Lecture Notes in Computer Science*, pages 358–375, Lisbon, Portugal, March 1998. Springer.
- [7] A. Podelski, W. Charatonik, and M. Müller. Set-based failure analysis for logic programs and concurrent constraint programs. In S. D. Swierstra, editor, *Programming Languages and Systems. Proceedings of the 8th European Symposium on Programming ESOP'99*, volume 1576 of *Lecture Notes in Computer Science*, pages 177–192, Amsterdam, The Netherlands, March 1999. Springer.

4.2 Deductive Model Checking

Investigators: Giorgio Delzanno, Andreas Podelski, Supratik Mukhopadhyay

In the second case of program analysis, we analyze programs over infinite numeric data, i.e. (unbounded) integers and reals. As in the first case of set-based analysis described in the section above, we translate the analysis problem to the problem of solving constraints overs sets, here sets of tuples of numbers. There are two important differences, however. (1) We do not weaken the analysis problem to a decidable one. This is simply because we don't know of any useful decidable version of the problem. (2) The solved form of the constraints is a non-recursive definition of the solution (expressing e.g. a union of polyhedra). This is because the machinery for dealing with recursive definitions is not available; its development is still part of our plans for future work. Until today, our method consists of eliminating all the 'recursion' that directly stems from the loops of the system to be analyzed. Since this method takes the same basic control structures (viz. fixpoint iterations) as the model checking method but is based not on exhaustive state space exploration but on deduction, we have coined the term deductive model checking for our general method.

In [5] we use the terminology and the formal setup of Constraint Logic Programming (CLP). We can write a constraint over sets of, say, reals as a special form of a constraint logic program with arithmetic constraints over the domain of reals. (We here employ the unusual view that predicates are free variables (over sets) of the logical formula that underlies a CLP program; note that 'CLP variables' (over reals) are all bound in this formula.) This allows us to exploit existing concepts and systems for programming over, say, arithmetic constraints as first-class data structures. In particular, the CLP-based setting has helped us to find optimizations (related to fixpoint evaluation strategies and to abstraction) that are natural, directly implementable and provably correct. Moreover, we have implemented a deductive model checking method in terms of transformations

of CLP programs. This implementation shows a competitive performance (on benchmark-like examples) thanks to the built-in constraint solver and other programming facilities of a CLP system (see also Section 10.4).

The work reported in [5] is interesting also from the perspective of the field of logic programming since it proposes a paradigm shift: instead of aiming at the *synthesis* of operational behavior from programs viewed as executable specifications, do the *analysis* of the operational behavior of given systems through the CLP programs obtained by a translation. The classical correspondence between denotational semantics and operational semantics becomes here, for the first time, useful. In [6], we further elaborates on this aspect.

We have used [5] as a starting point to investigate abstractions. In [2], we define an abstraction similar to widening that accelerate least fixpoint computations in model checking over integers. We show that this abstraction is complete, i.e., we do not lose precision by applying it. Preliminary experimental results indicate the potential usefulness of our abstraction techniques. In [3], we define a narrowing operator to accelerate greatest fixpoint computations in model checking for liveness properties of integer-valued systems. To our knowledge this is the first proposal of a narrowing operator in symbolic model checking. Our narrowing operator has the additional property that the resulting greatest fixpoint is accurate. Other recent work is on the verification of parameterized broadcast protocols [1] specified by integer-valued programs and on the relaxation of the constraint-solving problems over integers to the reals in model checkers for integer-valued programs [4].

We have applied our setup of constraints/CLP-programs also to the verification of timed systems. In [7], we introduce the subclass of Timed Logical Processes (TLP's) and establish their formal connection with the standard model, timed automata. We use this connection to explain the industrial-scale timed model checker UPPAAL in terms of XSB-style tabling with constraints. This allows us to directly obtain a competitive implementation of the corresponding model checking procedure in the CLP system of Sicstus Prolog, to enforce termination through an operation on constraints, and to increase the expressiveness of the underlying timed temporal logic.

References

[1] G. Delzanno, J. Esparza, and A. Podelski. Constraint-based analysis of broadcast protocols, May 1999. Submitted.

[2] G. Delzanno and A. Podelski. Complete Abstractions for Symbolic Model Checking, April 1999. Submitted for publication.

[3] G. Delzanno and A. Podelski. Model Checking Liveness Properties using Narrowing, April 1999. Submitted for publication.

[4] G. Delzanno and A. Podelski. Relaxation methods for symbolic model checking, May 1999. Submitted.

- [5] G. Delzanno and A. Podelski. Model Checking in CLP. In R. Cleaveland, editor, *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'99) held as part of ETAPS'99*, volume 1579 of *Lecture Notes in Computer Science*, pages 223–239, Amsterdam, The Netherlands, January 1999. European Association for Programming Languages and Systems (EAPLS); European Association for Theoretical Computer Science (EATCS), Springer.

- [6] G. Delzanno and A. Podelski. Verification of infinite-state systems in constraint logic programming. In F. Fages, editor, *Proceedings of the French Conference of Logic Programming*, March 1999. To appear.

[7] S. Mukhopadhyay and A. Podelski. Model checking for timed logical processes, May 1999. Submitted.

5 Numerical Constraints

The group on constraint programming (Alexander Bockmayr, Thomas Kasper and Friedrich Eisenbrand) has continued and considerably extended its work at the interface of finite domain constraint programming and integer linear programming. They have developed a unifying logical framework, branch-and-infer, that clarifies the relationship between these two approaches and shows how they can be integrated. Moreover, they have studied the complexity of Gomory-Chvátal cutting planes. They proved a polynomial upper bound on the Chvátal rank of 0/1 polytopes and solved a long-standing open question by showing that the membership problem for the first elementary closure is co-NP-complete.

5.1 Integration of integer linear programming and finite domain constraint programming

Investigators: Alexander Bockmayr and Thomas Kasper

Integer linear programming and finite domain constraint programming are two general approaches for solving discrete optimization problems. In order to clarify the relationship between these two approaches and to show how they can be integrated, we have developed a unifying logical framework, branch-and-infer [1, 2]. The framework is based on a distinction between primitive and non-primitive constraints. Primitive constraints are those constraints that can be solved easily and for which global methods are available. Non-primitive constraints are those constraints for which such methods do not exist and which make the problem hard to solve. In integer linear programming, the primitive constraints are linear equations and inequalities, which are solved over the real (or rational) numbers. The only non-primitive constraint is `integer`, i.e. the condition that some or all variables should take integer values. In finite domain constraint programming, the primitive constraints are domain constraints of the form $x \leq 2, y \geq 3, z \neq 4, x = y$, which are solved over the integer numbers. All other constraints are non-primitive. This includes more general arithmetic constraints, like linear equations, inequalities or disequalities in several variables, and symbolic constraints like `alldifferent` or `cumulative`.

Symbolic constraints are one of the main reasons for the success of constraint programming. On the one hand, they extend the constraint language and allow one to model many problems in a much more natural and compact way. For example, `alldifferent`$([x_1, \ldots, x_n])$ states that the variables $x_1, \ldots, x_n$ should take pairwise different values, which cannot be expressed easily by linear equations or inequalities. On the other hand, symbolic constraints make it possible to incorporate efficient algorithms for a specific problem area into a general solver. For example, to handle the `alldifferent` constraint, one can use the theory of matching in bipartite graphs. Thus, symbolic constraints not only increase the expressive power of the constraint language. But, they are also crucial for the efficiency of problem solving.

The basic idea underlying the branch-and-infer framework is that, in both integer linear programming and finite domain constraint programming, problems are solved by a combination of inference and search. The primitive constraints define a relaxation of the problem, for which an efficient global solution method is available. The non-primitive constraints are handled locally by an inference agent that derives from a given non-primitive constraint and the current relaxation new primitive constraints that tighten this relaxation. Since, in general, a problem cannot be solved using the relaxation alone, inference has to be combined with search, which together provide a complete solution method.

In integer linear programming, the primitive constraints are solved by linear programming meth-

ods, e.g. the Simplex algorithm. To handle the non-primitive constraint `integer`, general cutting plane techniques, e.g the Gomory-Chvátal method or disjunctive programming, can be applied as inference algorithms. In finite domain constraint programming, the non-primitive constraints are handled by local consistency algorithms that reduce the domain of the variables, which corresponds to the inference of new bound inequalities or disequalities in the branch-and-infer framework.

Branch-and-infer provides a sound theoretical basis for the integration of integer linear programming and finite domain constraint programming. In particular, it indicates how to introduce symbolic constraints into integer linear programming, where they can play a similar role as in constraint programming. Concerning expressiveness, symbolic constraints extend the language of linear equations and inequalities. Symbolic constraints allow the modeler to include large families of linear inequalities into the model, without writing them down explicitly. For example, when solving a traveling salesman problem, we might use a symbolic constraint `tsp`(...) to state the problem-defining degree and the subtour elimination constraints. Declaratively, this constraint is equivalent to exponentially many linear inequalities. Operationally, however, only some of these inequalities will be added to the model at runtime (as cutting planes). Another example would be a non-linear constraint in 0-1 variables. Declaratively, this defines a set X of 0-1 vectors and, by polarity, a set of linear inequalities valid for X. Operationally, we can realize such a constraint by a linearization procedure, which again will add to the model only some of the linear inequalities that are implied by the constraint.

Concerning efficiency, symbolic constraints allow one to integrate specialized cutting plane algorithms based on polyhedral combinatorics into a general branch-and-cut solver. Symbolic constraints give the modeler the possibility to identify some specific structure in the problem, which later can be exploited when the model is solved. For example, when we solve a model containing the symbolic constraint `tsp`, we can enhance our general branch-and-cut solver by computing specialized cutting planes for `tsp` instead of using more general cutting planes for arbitrary linear 0-1 programs.

References

- [1] A. Bockmayr and T. Kasper. Branch-and-infer: A unifying framework for integer and finite domain constraint programming. *INFORMS Journal on Computing*, 10(3):287–300, 1998.
- [2] T. Kasper. *A Unifying Logical Framework for Integer Linear Programming and Finite Domain Constraint Programming.* PhD thesis, Universität des Saarlandes, December 1998.

5.2 Elementary closure and bounds on the Chvátal rank of 0-1 polytopes

Investigators: Alexander Bockmayr and Friedrich Eisenbrand

The Gomory-Chvátal rounding procedure is a method for computing the integer hull of a polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m$. The procedure is the successive application of all Gomory-Chvátal cuts to P. The number of iterations needed to compute the integer hull P_I of P is called the Chvátal rank of the polyhedron P. This notion was introduced by Chvátal in 1973 as an indicator for the "degree of discreteness" and thus the complexity of an integer linear program of the form $\max\{c^T x \mid x \in P \cap \mathbb{Z}^n\}, c \in \mathbb{R}^n$. For a long time, a polynomial upper bound of the rank function for polytopes in the 0/1 cube was not known and other cutting planes based on the lift-and-project method have been argued to be superior to Gomory-Chvátal rounding, since the rank defined by these methods is at most the dimension of the cube.

In a recent paper (with M. Hartmann and A. S. Schulz) [2] (see also [1] for an earlier geometric proof of this result) we showed that the Chvátal rank of a polytope in the 0/1 cube is bounded

by a function in $O(n^3 \log n)$. This polynomial upper bound is achieved by scaling possibly large integral facet-defining vectors, i.e., using cutting planes defined by short integer normal vectors first and by postponing cuts with huge normal vectors until the end, when the relaxation is already tight enough. Gomory-Chvátal cuts can be used in a branch-and-cut framework. Our theoretical observations are compliant with observations made in practice, namely that cutting planes defined by short vectors are preferable.

In [4] we improved this bound to $O(n^2 \log n)$. We also showed that the rank of valid inequalities $c^T x \leq \delta$, where $\|c\|_\infty$ is bounded by a constant is at most $\mathrm{O}(n)$. The latter observation explains why for most cutting planes derived in polyhedral studies of several popular combinatorial optimization problems only linear growth has been observed; the coefficients of the corresponding inequalities are usually small.

In 1986, Schrijver asked whether there is a polynomial algorithm for the problem

> Given a matrix $A \in \mathbb{Z}^{m \times n}$, a vector $b \in \mathbb{Z}^m$ and a rational $x \in \mathbb{Q}^n$, is x outside of the elementary closure of $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$?

This question is motivated for example by the fractional matching polytope. If it was possible to optimize over the elementary closure of a polyhedron in polynomial time, then this would imply a polynomial algorithm for the weighted matching problem, since the fractional matching polytope has Chvátal rank 1. We have been able to provide a negative answer to this longstanding open question. In [3], we show that the problem mentioned above is NP-complete. By the equivalence of separation and optimization, it follows that the optimization problem over the elementary closure is NP-hard.

References

- [1] A. Bockmayr and F. Eisenbrand. On the Chvátal rank of polytopes in the 0/1 cube. Research Report MPI-I-97-2-009, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, September 1997.
- [2] A. Bockmayr, F. Eisenbrand, M. Hartmann, and A. S. Schulz. On the Chvátal rank of polytopes in the 0/1 cube. *Discrete Applied Mathematics*. To appear.
- [3] F. Eisenbrand. Note on the membership problem for the elementary closure of a polyhedron. *Combinatorica*. To appear.
- [4] F. Eisenbrand and A. S. Schulz. Bounds on the Chvátal rank of polytopes in the 0/1-cube. In *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, Graz, Austria, 1999. Springer. To appear.

6 Symbolic Constraints

Constraint satisfaction problems usually arise from logic modelling of certain application domains (e.g. performance analysis, computational linguistics), or as well-defined subproblems of logic deduction in general theorem proving. We speak of symbolic constraints if the interpretation of constraints is relative to structures over domains which are well-suited for the representation of symbolic expressions.

6.1 Set constraints

Investigators: Witold Charatonik, Andreas Podelski, Jean-Marc Talbot

Although work on set constraints already started with John Reynolds in 1969 and Neil Jones in 1979, the topic has received broad interest mainly in the nineties, when the term 'set constraints' was coined and the general problem of solving set constraints was formulated. Important results before 1997 include work by Heintze, Aiken, Tison, Kozen, Vardi and McAllester.

A well-known special case of set constraints are regular systems of equations over strings (systems which correspond to automata, viz. nfa's). In more general set constraints, one goes from strings to trees and one adds more set operations. The motivation to consider sets of trees stems from the fact that trees model symbolic information and are thus used in all tools for reasoning about programs (ranging from specific tools such as compilers or verifiers to general ones such as logic or functional programming systems). Set constraint solvers are employed as enhancements of such tools on various levels (e.g. for pruning proof search, or for inferring types) and can be viewed as a program reasoning tool on its own; see also Section 4. In [13, 14] we give overviews of this area.

We have continued to pursue the research direction that we have already outlined in the Third Biennial Report of MPI-I. Namely, given the NEXPTIME-completeness results of the general class of set constraints, it is worth to single out practically relevant and fundamentally interesting classes with a restricted expressiveness and investigate their (hopefully better) algorithmic properties. We found that there is an important gap: it seems that for many classes, adding expressiveness pushes the time complexity in one step from cubic to DEXPTIME-complete. This tells us in a drastic way that the extra precision that one obtains in the corresponding program analyses has its price.

Jones 1979 introduced atomic set constraints for the analysis of while programs over trees. In [17], an extended version of [10], we propose a constraint system with atomic constraints that are, however, interpreted over non-empty sets of finite trees. We give an $O(n^3)$ time algorithm for testing satisfiability, the first incremental one in this area. This work has been picked up by Fähndrich at Microsoft Research; at a Dagstuhl seminar on program analysis in April 1999 he presented an implementation of the algorithm for large-scale problems, with important (not always sound) optimizations.

In [11] we introduce the constraint system $\text{FEAT}_{\leq}$ and investigate its logical properties and expressiveness. The constraints are essentially atomic set constraints interpreted over feature trees (instead of standard Herbrand trees. The satisfiability notion does not change whether we take feature trees (with the subtree relation) or non-empty sets of feature trees (with the inclusion) for the interpretation. We solve the satisfiability and entailment problems of $\text{FEAT}_{\leq}$ in cubic time and prove that $\text{FEAT}_{\leq}$ has the independence property. A revised version [12] of [11] will appear in a special issue of *Constraints*.

Definite set constraints arise from atomic ones by adding the intersection operator. This addition is motivated either by the gain of precision in Jones' analysis or by the analysis of programs with

alternation, to which logic programs or concurrent constraint programs belong. These constraints form the historically first class for which the decidability was shown (by Heintze/Jaffar 1991). In [6, 9] we introduce a natural class that we simply call set constraints with intersection and show that its satisfiability problem is DEXPTIME-complete. We prove the equivalence with definite set constraints, and thereby settle the complexity question also for that class. The complexity characterization continues to hold when we add negated inclusions or when we consider entailment instead of satisfiability.

In [8, 7] we introduce the class of co-definite set constraints. This class is motivated by the analysis of the kind of programs mentioned above, but wrt. liveness properties such as termination, or for the type analysis wrt. ongoing program behavior. We show that its satisfiability problem is DEXPTIME-complete. The duality between definite and co-definite set constraints lies in the existence of least resp. greatest solutions.

Both, the classes of definite and co-definite set constraints are further investigated in [15]. We extend these two classes by adding an intensional set construction, the so called membership expression. As we can prove, the extensions strictly increase the expressiveness but preserve the properties of the existence of the least or greatest solutions and of the DEXPTIME-completeness for the satisfiability problem.

It turns out that there is an interesting difference in complexity between the case of a finite signature for the tree algebra and the case of an infinite one. In [16] we investigate entailment problem for the class of atomic set constraints; we show that it can express the validity problem of quantified boolean formulas and is thus PSPACE-hard. For infinite signatures, we also present a PSPACE-algorithm for solving atomic set constraints with negation. This proves that entailment of atomic set constraints is PSPACE-complete for infinite signatures. In case of finite signatures, the problem is known to be DEXPTIME-hard.

We continued to investigate the frontier between decidable and undecidable classes of set constraints. In [4] we prove the undecidability of the $\exists^*\forall^*$-fragment of the first-order theory of atomic set constraints. In [3] we studied possible extensions of set constraints by adding additional equational axioms like associativity or commutativity. It turns out that in the most interesting cases (associativity, associativity together with commutativity) the satisfiability problem becomes undecidable.

We have succeeded to improve the complicated solutions for two hard problems that we had given previously. In [5] we investigate the classes of positive as well as positive and negative set constraints. These are the two NEXPTIME-complete classes that attracted the most attention in the past. We present a new simple algorithm for testing satisfiability, based on a new class of tree automata that we introduce. The ideas from this paper are extended in [2] to give a new solution to the satisfiability problem for the general class of set constraints; this solution will be understood by more people than the previous one in [1].

References

[1] W. Charatonik and L. Pacholski. Set constraints with projections are in NEXPTIME. In *Proceedings of the 35^{th} Symposium on Foundations of Computer Science*, pages 642–653, 1994.

[2] W. Charatonik and L. Pacholski. Set constraints. In preparation, 1999.

- [3] W. Charatonik. Set constraints in some equational theories. *Information and Computation*, 142:40–75, 1998.
- [4] W. Charatonik. An undecidable fragment of the theory of set constraints. *Information Processing Letters*, 68:147–151, 1998.

- [5] W. Charatonik. Automata on dag representations of finite trees. Research Report MPI-I-1999-2-001, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, March 1999.
- [6] W. Charatonik and A. Podelski. Set constraints with intersection. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97)*, pages 362–372, Warsaw, Poland, 1997. IEEE Comput. Soc. Press.
- [7] W. Charatonik and A. Podelski. Solving set constraints for greatest models. Research Report MPI-I-97-2-004, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, November 1997.
- [8] W. Charatonik and A. Podelski. Co-definite set constraints. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 211–225, Tsukuba, Japan, March 1998. Springer.
- [9] W. Charatonik and A. Podelski. Set constraints with intersection. *Information and Computation*, 1999. To appear.
- [10] M. Müller, J. Niehren, and A. Podelski. Inclusion constraints over non-empty sets of trees. In M. Bidoit and M. Dauchet, editors, *Proceedings of the 7th International Joint Conference CAAP/FASE: Theory and practice of software development (TAPSOFT-97)*, volume 1214 of *Lecture Notes in Computer Science*, pages 345–356, Lille, France, April 1997. Springer.
- [11] M. Müller, J. Niehren, and A. Podelski. Ordering constraints over feature trees. In G. Smolka, editor, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP-97)*, volume 1330 of *Lecture Notes in Computer Science*, pages 549–562, Linz, Austria, 1997. Springer.
- [12] M. Müller, J. Niehren, and A. Podelski. Ordering constraints over feature trees. *Constraints*, 1999. To appear.
- [13] L. Pacholski and A. Podelski. Set constraints: a pearl in research on constraints. In G. Smolka, editor, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP-97)*, volume 1330 of *Lecture Notes in Computer Science*, pages 549–562, Linz, Austria, 1997. Springer.
- [14] A. Podelski. Set-based analysis of logic programs and reactive logic programs. In J. Maluszynski, editor, *Proceedings of the International Symposium on Logic Programming (ILPS-14)*, pages 35–36, Leuven, Belgium, 1997. MIT Press.
- [15] J.-M. Talbot, P. Devienne, and S. Tison. Generalized definite set constraints. *CONSTRAINTS - An International Journal*, 5, 1999. To appear.
- [16] J.-M. Talbot, J. Niehren, and M. Müller. Entailment of atomic set constraints is pspace-complete. In G. Longo, editor, *Proceedings of the 14th Annual IEEE Symposium on logic in computer science (LICS'99)*, Trento, Italy, 1999. IEEE Comput. Soc. Press. To appear.

  [17] M. Müller, J. Niehren, and A. Podelski. Nonempty set constraints. Extended version of [10], 1999.

6.2 Subtype Constraints

Investigators: Sergei Vorobyov

Solving systems of subtype constraints (or subtype inequalities) is in the core of efficient type reconstruction in modern object-oriented languages with subtyping and inheritance, two problems known polynomial time equivalent. It is important to know how different combinations of type constructors involved in the underlying type system influence the complexity of the problem. Different classes of subtype constraints in various type systems have been well studied in the literature during the last decade.

In [6, 3] we demonstrated the NP-hardness and NEXPTIME-decidability of the satisfiability problem for subtype inequalities between object types built by using simultaneously the functional and the nonempty record type constructors, but without any atomic types and atomic subtyping.

The class of constraints we address naturally arises in object-oriented programming and is intermediate with respect to known classes. For pure functional types with atomic subtyping of a special non-lattice (crown) form solving subtype constraints is PSPACE-complete [5, 1]. On the other hand, if there are no atomic types and subtyping on them, but the largest $\top$ type is included, then both pure functional and pure record (separately) subtype constraints are polynomial time solvable [2, 4], which is mainly due to the lattice type structure. We showed that combining the functional and nonempty record constructors yields NP-hardness without any atomic subtyping, and the same is true for just a single type constant with a nonempty record constructor.

References

[1] A. Frey. Satisfying subtype inequalities in polynomial space. In *Static Analysis (SAS'97)*, volume 1302 of *Springer Lecture Notes in Computer Science*, pages 265–277, 1997.

[2] D. Kozen, J. Palsberg, and M. I. Schwartzbach. Efficient inference of partial types. *J. Comput. Syst. Sci.*, 49:306–324, 1994.

- [3] S. Vorobyov. Subtyping functional+nonempty record types. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic (CSL'98)*, volume 1584 of *Lecture Notes in Computer Science*, pages 280–295, Brno, Czech Republic, 1999. European Association for Computer Science Logic, Springer.

[4] J. Palsberg. Efficient inference of object types. *Information and Computation*, 123:198–209, 1995. Preliminary version in LICS'94.

[5] J. Tiuryn. Subtype inequalities. In *Logic in Computer Science (LICS'92)*, pages 308–315, 1992.

[6] S. Vorobyov. Satisfiability of functional+record subtype constraints is NP-hard. Research Report MPI-I-98-2-004, Max-Planck Institut für Informatik, Im Stadtwald, Saarbrücken, D-66123, Germany, January 1998.

6.3 Quantified Theories of One Step Rewriting

Investigators: Sergei Vorobyov

A finite term rewriting system R generates the binary one step reducibility relation $\mathbf{R}$ on the set of ground terms. A theory of one step rewriting in R is the first-order theory of this binary relation $\mathbf{R}$ formulated in the language of the predicate calculus without equality containing the unique binary predicate symbol R interpreted as $\mathbf{R}$. The problem whether first-order theories of one step rewriting in finite systems are decidable was several times iterated in the RTA list of open problems.

The motivation for the problem is quite natural. For example, the ground reducibility of a term and the strong confluence of a system are expressible by the formulas of the theory. Note that both properties are known to be decidable. Similarly, the decidability of properties like encompassment, known to be decidable, would follow from the general decidability of theories of one step rewriting. Recall also that the first-order theories of one step rewriting in finite ground systems are decidable. On the other hand, the transitive closure of the one step reducibility relation seems to be inexpressible in the theories of one step rewriting (the opposite would immediately lead to their undecidability). All these facts motivated the quest for the solution to the above problem and for the general decision procedure applicable to all rewrite systems. This would have allowed

to decide all properties of rewrite systems, like discussed above, expressible in the language of one step rewriting uniformly.

Unfortunately, the problem appeared undecidable, but the existing proofs suggested that non-termination of a system was essential. Therefore the question remained concerning the existence of finite finitely terminating systems, possibly linear, that have (un)decidable theories of one step rewriting. The similar decidability problem was put forward for the subclass of linear systems.

The decidability conjectures for finitely terminating and linear systems were first dispelled in [1], where a fixed finite, simultaneously finitely terminating and linear system with undecidable theory of one step rewriting was constructed. The proof was given by reduction from the theory of binary concatenation (finitely generated free semigroups), well known to be undecidable. As a practical drawback compensating for the ease of reduction, the quantifier alternation of the sentences forming the undecidable class was quite high.

In [2] we further improved and sharpened the above undecidability results by showing that no decision algorithm can decide the $\exists\forall\forall\forall$-theory of any given finite, simultaneously 1) finitely terminating, 2) linear, and 3) confluent rewrite system. Namely, by reduction from the halting problem for Minsky's two-register machines in [2] we proved that there is no algorithm capable of deciding the $\exists\forall\forall\forall$-theory of one step rewriting of an arbitrary finite linear confluent finitely terminating term rewriting system (weak undecidability). We also present a fixed such system with undecidable $\exists\forall^*$-theory of one step rewriting (strong undecidability). This improves over all previously known results of the same kind. All the preceding proofs constructed non-confluent systems and used more complicated quantifier prefixes.

We also construct a fixed finite linear canonical system with undecidable $\exists\forall^*$-theory of one step rewriting (strong undecidability). It is also important to note that the weak undecidability results of all other authors do not imply the existence of such systems.

References

- [1] S. Vorobyov. The first-order theory of one step rewriting in linear noetherian systems is undecidable. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications (RTA-97)*, volume 1232 of *Lecture Notes in Computer Science*, pages 254–268, Sitges, Spain, June 1997. Springer.

[2] S. Vorobyov. The undecidability of the first-order theories of one step rewriting in linear canonical systems. Research Report MPI-I-98-2-009, Max-Planck Institut für Informatik, May 1998.

6.4 Second-Order Unification, Rigid Reachability, and Rigid E-Unification

Investigators: Veronique Cortier, Harald Ganzinger, Florent Jacquemard, Margus Veanes, Sergei Vorobyov, Andrei Voronkov

Second-order unification or SOU generalizes first-order unification by allowing variables to occur also in the position of function symbols, and is undecidable in general [4]. For example, the second-order terms $G(f(c,c))$ and $f(f(c,c),G(c))$ (with the second-order variable G) have a unifier that maps G to a λ-term $\lambda x.f(x,x)$. Application of that unifier to either term yields the term $f(f(c,c),f(c,c))$. Second-order unification and restricted forms thereof (such as context unification, that appears as a subproblem in constraint solving with membership constraints and distributive unification), play a fundamental role in several areas.

Rigid reachability is the problem, given a rewrite system R and two terms s and t, to decide if there exists a ground substitution σ such that $s\sigma$ rewrites in some number of steps via $R\sigma$ into

$t\sigma$. Simultaneous rigid reachability or SRR is the problem in which a substitution is sought which simultaneously solves each member of a system of reachability constraints (R_i, s_i, t_i). A special case of SRR arises when the R_i are symmetric, i.e., containing for each rule $l \to r$ also its converse $r \to l$. The latter problem was introduced in [3] as simultaneous rigid E-unification or SREU. It has been shown in [1] that SREU is undecidable, whereas the non-simultaneous case (with just one constraint) is NP-complete [2]. An overview of methods of equality reasoning in sequent-based systems is presented in [9], including the history of handling equality in sequent systems, methods based on rigid E-unification, paramodulation-based methods, the equality elimination method, and equality reasoning in non-classical logics.

In [13] the connection between SOU and SREU is studied in detail. These problems are shown to be almost the same and this connection is used to give a very elementary undecidability proof of SOU, improving a resent result in [16]. This connection shows also that SOU is closely related to constraint satisfaction problems that arise in tableaux based global proof search in logic with equality. In [12] the undecidability of SOU is studied further for various restricted fragments of SOU. For example, it is shown that SOU is undecidable already with a single unary second-order variable.

In [10] it is shown that for (non-symmetric) rigid reachability already the case of a single reachability constraint is undecidable, even when the rule set is ground. The main tool in that proof is the Shifted Pairing Theorem from [11]. From this follows the undecidability of a rather restricted form of SOU for problems which contain just a single second-order variable which, in addition, occurs at most twice in the unification problem.

In [7] some restricted fragments of SRR are shown to be decidable. The main results are that monadic SRR with ground rules is PSPACE-complete, and that balanced SRR with ground rules is EXPTIME-complete. The first result indicates the difference in computational power between fragments of SREU with ground rules and non-ground rules an improves some results in [5]. The second result improves some results in [8]. In [8] it is proved that SREU with one variable is decidable and even P-complete when the number of reachability constraints is fixed. Also, some fragments of SREU with more than one variable are shown to be decidable.

Context unification is a particular case of second-order unification in which all second-order variables are unary and only linear functions are admitted as solutions. Context unification is useful in different areas of Computer Science: term rewriting, theorem proving, equational unification, constraint solving, computational linguistics, software engineering. Its decidability is an intriguing open problem, with only a very poor known NP-lower bound. The problem is know to be difficult as a proof of decidability would extend the famous result by [6] according to which the solvability of equations in a free semigroup is decidable: context unification coincides with word unification in the case of monadic signatures, where function symbols have arity of at most one.

In [14, 15, 17] we have presented a series of results which gradually strengthen undecidability results about quantified fragment of the theory of context unification. In the end we were able to show that the set of $\forall\exists^5$-quantified context equations (i.e., sentences of the form $\forall W\ \exists\, U, V, S, G, H\ \ s = t$) is undecidable and, in fact, is co-recursively enumerable hard (i.e., every set with recursively enumerable complement is many-one reducible to it).

References

[1] A. Degtyarev and A. Voronkov. Simultaneous rigid E-unification is undecidable. UPMAIL Technical Report 105, Uppsala University, Computing Science Department, May 1995.

[2] J. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Rigid E-unification is NP-complete. In *Proc.*

IEEE Conference on Logic in Computer Science (LICS), pages 338–346. IEEE Computer Society Press, 1988.

[3] J. Gallier, S. Raatz, and W. Snyder. Theorem proving using rigid E-unification: Equational matings. In *Proc. IEEE Conference on Logic in Computer Science (LICS)*, pages 338–346. IEEE Computer Society Press, 1987.

[4] W. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[5] Y. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. In P. Degano, R. Corrieri, and A. Marchetti-Spaccamella, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 154–165. Springer, 1997.

[6] G. Makanin. The problem of solvability of equations in free semigroups. *Mat. Sbornik (in Russian)*, 103(2):147–236, 1977. English Translation in American Mathematical Soc. Translations (2), vol. 117, 1981.

- [7] V. Cortier, H. Ganzinger, F. Jacquemard, and M. Veanes. Decidable fragments of simultaneous rigid reachability. In *Proc. ICALP'99*, 1999. To appear. Full version of this paper is available as MPI Research Report MPI-I-1999-2-004.

- [8] A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. Decidability and complexity of simultaneous rigid e-unification with one variable and related results. *Theoretical Computer Science*, 1999. To appear.

- [9] A. Degtyarev and A. Voronkov. Equality reasoning in sequent-based calculi. Research Report MPI-I-98-2-011, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, July 1998.

- [10] H. Ganzinger, F. Jacquemard, and M. Veanes. Rigid reachability. In J. Hsiang and A. Ohori, editors, *Proceedings of the 4th Asian Computing Science Conference on Advances in Computing Science (ASIAN-98)*, volume 1538 of *Lecture Notes in Computer Science*, pages 4–21, Manila, The Philippines,, 1998. Springer. A full version of this paper is available as MPI-I Research Report MPI-I-98-2-013.

- [11] Y. Gurevich and M. Veanes. Logic with equality: Partisan corroboration, and shifted pairing. *Information and Computation*, 1999. To appear.

- [12] J. Levy and M. Veanes. On the undecidability of second-order unification. *Information and Computation*, 1999. To appear.

- [13] M. Veanes. The relation between second-order unification and simultaneous rigid e-unification. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 264–275, Indianapolis, Indiana, 1998. IEEE Computer Society Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.

- [14] S. Vorobyov. $\forall\exists^*$-equational theory of context unification is Π^0_1-hard. Research Report MPI-I-98-2-008, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, April 1998. Revised and abridged version appeared in the Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98), Springer Lecture Notes in Computer Science. Brno, Czech Republic, August 24-28.

- [15] S. Vorobyov. $\forall\exists^*$-equational theory of context unification is Π^0_1-hard. In L. Brim, J. Gruska, and J. Zlatuska, editors, *Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS-98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 597–606, Brno, Czech Republic, August 1998. Mazaryk University, European Assoc of Computer Science Logic, Springer.

[16] A. Schubert. Second-order unification and type inference for Church-style polymorphism. In *Conference Record of POPL'98: The 25TH ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 279–288, San Diego, California, January 1998. ACM Press.

[17] S. Vorobyov. $\forall\exists^5$-equational theory of context unification is undecidable. Submitted, improving the result from the preliminary version in MFCS'98, LNCS 1450, 1999.

7 Extended modal logic and automated theorem proving

Modal logics and its relatives play a crucial role in various fields of computer science including temporal reasoning, knowledge representation, and logics of programs. Their popularity probably is due mainly to their natural possible worlds semantics and their decidability (in many cases). It is well-known that propositional modal logics can be embedded into first-order logic using a translation morphism based on the Kripke semantics for these logics. However, although the mapping itself preserves the properties of the modal logics and modal formulae under consideration, the application of standard theorem proving procedures and calculi for first-order logic does not. In particular, the decidability of propositional modal logics is easily lost. Therefore, it is important to investigate alternative embeddings into first-order logic as well as refinements of existing calculi for first-order logic which provide better means for obtaining decision procedures for modal logics in the context of first-order logic.

In section 7.1 we describe some of our general lines of investigation into translations of non-classical into classical logics, bringing to completion some of these lines. Section 7.2 follows with a more detailed presentation of our newest results on the functional translation approach which avoids many of the disadvantages mentioned above. One of the nice features of this method is that it preserves a lot of the original structure. However, it requires a certain equational background theory to be included which depends on the modal logic under consideration. Fortunately, the particular structure of these background theories can be cast into special inference rules based on theory-unification and theory-resolution as described in section 7.3. This approach turned out to have a significant impact on efficiency as the empirical study in section 7.4 shows.

Examining and generalizing the (relational) translation from modal logics to first-order predicate logic has lead to an interesting general syntactic pattern, the so-called guarded fragment of first-order logic, which also enjoys decidability and which nevertheless is fairly expressive. Our work on these and related issues is described in section 7.5.

However, it sometimes turns out that the syntactical and semantical features of modal logics are too restrictive for certain purposes. We therefore also examined some modal logic extensions, the so-called hybrid logics. In section 7.6 we present our newest results on this issue.

7.1 Translations from modal into classical logic

Investigator: Andreas Nonnengart

Non-classical logics have been developed to describe human-oriented applications. Some of them extend classical logic with additional operators, connectives, and quantifiers for applications involving time, knowledge, belief, necessity, actions, etc. Others even change the basic deductive structure of classical logic, as, for instance, in intuitionistic logic, relevance logic, linear logic, paraconsistent logic, many-valued logic or fuzzy logic.

Nevertheless, classical logic has remained the main workhorse of logic and its applications, and for good reasons: It has very well understood and well-developed computational aspects and it is expressive in the sense that one can translate into it (especially into higher-order classical logic) most of the non-classical logics we might be interested in. The idea, therefore, is to encode non-classical logics in classical logic in a way that preserves the most prominent syntactical and semantical features of the encoded logic.

In [3] we present various ways of encoding (axiomatizing, translating) a logic in predicate logic in such a way that predicate logic theorem provers, in particular first-order predicate logic theorem provers, can be used to reason about the given logic. Here we examined various such methods. A syntactic translation, for instance, would be to encode a Hilbert-style consequence relation in

first-order logic. Semantic translations, on the other hand, usually encode semantic information of the logic under consideration, e.g. accessibility relations between possible worlds. In fact, we emphasized on translations based on possible worlds semantics. To this end we considered standard translations of normal propositional modal logics, of intuitionistic and relevance logic as well as quantified modal logics. But not only standard translations have been investigated. We also looked for alternative encodings that arise from related, yet slightly different semantics for the non-classical logics we were interested in. Typical examples can be found in the functional and in the semi-functional translation approaches. Finally, we also examined the possibilities to perform indirect translations. As an example consider the provability logic *Grz* (from Grzegorczyk). Its frames are not first-order describable; nevertheless it is possible to translate *Grz*-formulae into S4-formulae such that the translation is an S4-theorem if and only if the original formula is a *Grz*-theorem. For S4, on the other hand, our encoding in classical logics applies.

It turns out that quite often there are different ways of encoding a logic in predicate logic, and predicate logic theorem provers behave differently for different encodings. This opens possibilities to tune the encoding and to optimize the efficiency of predicate logic theorem provers for reasoning with encoded non-classical logic formulae.

One such tuning, for instance, can be found for modal logics after realizing that background theories that descend from the modal logic's peculiarities are not unique. This ambiguity in general is due to the syntactic restrictions on modal logics. It is thus of interest not only to find some suitable background theory, but find the simplest one. For example, it is known that the modal logic S5 is characterized by reflexivity, symmetry, and transitivity of the underlying accessibility relation. Nevertheless, it can be shown that S5 is also characterized by the more general universal relation. Doubtless, it is easier for a theorem prover to work with the universal relation than with equivalence relations. Therefore, such frame simplifications are always worth being detected. In [2] we describe our method of finding such simplifications with the help of auxiliary modalities. The main idea is based on the observation that we often can conservatively extend the syntax of the modal logic under consideration without changing the set of valid formulae that are describable in the original syntax. Translating this enriched logic into classical logic then reveals formulae that also talk about symbols that are introduced solely by the conservative extension. Eliminating these extra symbols with the help of a second-order quantifier elimination approach finally leads to a background theory which is often more general than the theories we would obtain if we had not considered the conservative extension.

As a tool for such a second-order quantifier elimination we introduced two methods: the SCAN algorithm [1, 6] by Hans Jürgen Ohlbach and Dov Gabbay and the Fixpoint approach developed in a joint work by A. Nonnengart and A. Szałas [5]. For an overview on both approaches, the SCAN algorithm and the Fixpoint approach, see [4].

References

[1] D. M. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992.

- [2] A. Nonnengart. Modal frame characterization by way of auxiliary modalities. *Logic Journal of the IGPL*, 6(6):875–899, 1998.
- [3] A. Nonnengart, H. J. Ohlbach, and D. Gabbay. Encoding two-valued non-classical logics in classic logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, Netherlands, 1999. To appear.
- [4] A. Nonnengart, H. J. Ohlbach, and A. Szalas. Quantifier elimination for second-order predicate logic.

To appear in *Logic, Language and Reasoning. Essays in honour of Dov Gabbay, Part I, Kluwer Academic Press*, 1999.

- [5] A. Nonnengart and A. Szalas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In E. Orlowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*, page 18. Springer, Berlin, 1999.

[6] H. J. Ohlbach. SCAN-elimination of predicate quantifiers. In M. A. McRobbie and J. K. Slaney, editors, *Proceedings of the 13th International Conference on Automated Deduction (CADE-13)*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 161–165, New Brunswick, New Jersey, USA, 1996. AT&T, Bell Labs, Lucent, IEEE, CADE inc, Springer.

7.2 An optimized translation method for modal logics and resolution

Investigator: Renate A. Schmidt

The thesis of Schmidt [3] studies and formalizes the optimized functional translation method for propositional modal logics and the treatment of inference in resolution procedures, in particular, theory resolution procedures. The optimized functional translation method follows the functional semantics of modal logic that defines accessibility between worlds by functions. A certain optimization is possible for propositional modal formulae [1, 2]. The optimization allows for universal and existential quantifiers to be swapped arbitrarily. In the relational context this operation is not admissible. However, in maximal or patched functional models swapping quantifiers preserves satisfiability. This property hinges on the generated frame property that embodies the fact that truth in a world of a modal formula does not depend on predecessor worlds.

The quantifier exchange operation is important for our decidability result, for it eliminates in the clausal forms all Skolem functions other than Skolem constants [4, 5]. Modal logics transform by the optimized functional translation to a lattice of clausal logics, called path logics. The weakest path logic is called basic path logic and is associated with the basic modal logics K and KD. It forms a fragment of monadic first-order logic with constant symbols and one binary function symbol (which can be embedded into the Bernay-Schönfinkel class). In [3] we show that resolution and condensing without additional refinement strategies is a decision procedure for basic path logic and certain of its extensions. This result is important for a number of reasons. One, unrefined resolution and condensing provides a decision procedure for the translation of many propositional modal logics, including K and arbitrary extensions with D, T and B, as well as their multi-modal versions, and also $S5$. Two, any resolution procedure with condensing and any compatible refinement strategy is a decision procedure for the relevant modal and path logics. For practical purposes this is paramount, since any fair implementation of a resolution theorem prover can serve as a reasonable and efficient inference tool for doing basic modal reasoning. This is confirmed by a series of benchmarks done with SPASS and other special purpose theorem provers (see section 7.4). Three, from a logical perspective, basic path logic appears to be the first solvable class (that is non-trivial) for which unrefined resolution and condensing solve the class.

The optimized functional translation method applies not only to modal formulae, but also to axiom schemas [2, 3]. A pleasant consequence is that some modal logics not determined by any elementary class of frames can be embedded in first-order logic. This extends the applicability for the resolution method (and other first-order theorem proving techniques) to essentially second-order modal logics, like K extended with McKinsey's schema. We make use of the new possibilities in a case study of accommodating reasoning in graded modal logic in a first-order resolution calculus enhanced with routines for doing addition and subtraction of integers [6]. Graded modal logic is

important in many applications, especially in knowledge representation and computational linguistics, because it includes numerical quantifiers such as 'there are at least n' or 'there are more than n'.

References

[1] A. Herzig. *Raisonnement automatique en logique modale et algorithmes d'unification.* PhD thesis, Univ. Paul-Sabatier, Toulouse, 1989.

- [2] H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *Journal of Logic and Computation*, 7(5):581–603, 1997. Also available as Research Report MPI-I-95-2-002.
- [3] R. A. Schmidt. *Optimised Modal Translation and Resolution.* PhD thesis, Universität des Saarlandes, November 1997.
- [4] R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, volume 87 of *CSLI Lecture Notes*, chapter 13, pages 189–208. CSLI, Stanford, USA, 1998.
- [5] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.

[6] H. J. Ohlbach, R. A. Schmidt, and U. Hustadt. Translating graded modalities into predicate logic. In H. Wansing, editor, *Proof Theory of Modal Logic*, volume 2 of *Applied Logic Series*, pages 253–291. Kluwer, 1996. Also available as Research Report MPI-I-95-2-008, Max-Planck-Institut für Informatik, Saarbrücken, Germany (May 1995).

7.3 E-unification and E-resolution for path logics

Investigator: Renate A. Schmidt

The optimized functional translation of modal logics induces a lattice of clausal logics, called path logics (see section 7.2). Different path logics are distinguished by different theories involving equations. The theories associated with serial modal logics consist exclusively of equations. Accordingly, equational reasoning needs to be done, and [5] studies theory resolution, in particular E-resolution, for path logics.

For the application to path logics previously introduced E-resolution calculi [7, 1] are too weak, for we need further refinements in order to force termination for decidable path logics. In [5] we show that the resolution framework of Bachmair and Ganzinger [4] can be adapted to provide a general calculus of ordered E-resolution with selection and an abstract notion of redundancy, which easily accommodates simplification rules required for different path theories. In combination with the method of renaming, ordered E-resolution results in a considerable efficiency gain.

Due to the characteristic properties of terms in path logics the unification problems are easier than for general terms. [6, 5] presents a formal treatment of E-unification and normalization for path theories explaining the core issues exemplified for the equations corresponding to the modal schemas T and 4, thus covering the unification problems for the modal logics *K*, *KD*, *KT*, *KD4*, *S4* and *S5*. Our algorithm combines adaptations of the general mutation rules considered separately in [2] and [3] for our forms of the identity law and the associativity law. Mutation rules have the advantage that paramodulating into terms can be avoided. Related unification algorithms and resolution calculi found in the literature are all designed for the non-optimized translations which require extended (strong) forms of Skolemization in order that a particular ordering within terms is preserved. Accordingly, our unification algorithms are more elegant and the proofs are considerably

simpler, though remaining technical. Our treatment pays special attention to normalization, which is essential.

References

[1] P. Baumgartner. An ordered theory resolution calculus. In A. Voronkov, editor, *Proc. LPAR'92*, volume 624 of *LNAI*, pages 119–130. Springer, 1992.

[2] H. Comon, M. Haberstrau, and J.-P. Jouannaud. Syntacticness, cycle-syntacticness and shallow theories. *Information and Computation*, 111(1):154–191, 1994.

[3] C. Kirchner and F. Klay. Syntactic theories and unification. In J. C. Mitchell, editor, *Proc. LICS'90*, pages 270–277, Philadelphia, 1990. IEEE Computer Society Press.

- [4] L. Bachmair and H. Ganzinger. A theory of resolution. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, 1999. To appear.

- [5] R. A. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, November 1997.

- [6] R. A. Schmidt. E-unification for subsystems of S4. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 106–120, Tsukuba, Japan, 1998. Springer.

[7] G. Plotkin. Building-in equational theories. In B. Meltzer and D. Michie, editors, *Machine Intelligence 7*, pages 73–90. American Elsevier, New York, 1972.

7.4 Empirical Evaluation of Modal Theorem Provers

Investigators: Ullrich Hustadt and Renate A. Schmidt

There have not been any exhaustive empirical evaluations or comparisons of the computational behaviour of modal theorem provers. We have conducted an extensive series of empirical performance test of a number of modal theorem provers on benchmark suites of randomly generated formulae [3, 4, 6]. The theorem provers tested are the Davis-Putnam-based procedure KSAT, the tableaux-based system $\mathcal{KRIS}$, the sequent-based Logics Workbench, and the optimized functional translation approach (discussed in section 7.2) combined with the first-order theorem prover SPASS [7].

Our benchmark suites are sets of multi-modal formulae in a certain normal form randomly generated according to the scheme of [1, 2]. We investigate the quality of the random modal formulae and show that the scheme has some shortcomings, which may lead to mistaken conclusions. We propose improvements to the evaluation method and show that our translation approach provides a viable alternative to the other approaches.

In [5] we study various schemes for enhancing the performance of modal tableau procedures. We discuss techniques and strategies for dealing with the nondeterminism in tableau calculi. We focus on two techniques which we think are key techniques for efficient modal tableau procedures, and indeed all modal decision procedures: Simplification, backjumping and dependency-directed backtracking. These techniques are well-known from other areas of computer science, like automated theorem proving in propositional logic, (constraint) logic programming and games, and deserve much more attention in the area of modal tableau theorem proving. Benchmark results obtained with randomly generated modal formulae show the effect of combinations of the different schemes.

References

[1] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures: The case study of modal K. In M. A. McRobbie and J. K. Slaney, editors, *Automated Deduction: CADE-13*, volume 1104 of *LNAI*, pages 583–597. Springer, 1996.

[2] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for $\mathcal{ALC}$. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proc. KR'96*, pages 304–314. Morgan-Kaufmann, 1996.

- [3] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logic. Research Report MPI-I-97-2-003, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, February 1997.
- [4] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logics. In M. E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 202–207, Nagoya, Japan, 1997. International Joint Conferences on Artificial Intelligence, Inc. (IJCAII) and Japanese Society for Artificial Intelligence (JSAI), Morgan Kaufmann.
- [5] U. Hustadt and R. A. Schmidt. Simplification and backjumping in modal tableau. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reaso ning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 187–201, Oisterwijk, The Netherlands, May 1998. Springer.
- [6] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, page 42, 1999. To appear.
- [7] C. Weidenbach. SPASS version 0.49. *Journal of Automated Reasoning*, 18(2):247–252, 1997.

7.5 Guarded Fragment

Investigators: Harald Ganzinger, Christoph Meyer, and Margus Veanes

One important motivation for the identification of decidable fragments of first-order logic is to explain the "robust" decidability [5] of (propositional) modal logics through the existence of a first-order fragment which is both, decidable and sufficiently expressive to capture a predominantly portion of modal logics. The guarded fragment GF [1] (cf. section 3.2 as well as the two-variable fragment FO^2 are prominent classes with, however, particular limits in both decidability and expressiveness. In [3] we have studied certain variants of the two-variable guarded fragment GF^2 of first-order logic (with equality) which corresponds to the intersection $\mathrm{GF} \cap \mathrm{FO}^2$ of the guarded fragment and the two-variable fragment. We mean, by variants of GF^2, the extension of formulae in GF^2 with certain constraints which impose additional closure properties on the relations, e.g. the transitive closure of a relation. By GF^2_-, we explicitly denote the two-variable guarded fragment without equality.

The two-variable fragment of first-order logic is decidable and refers to the class of all closed first-order formulae containing at most two variables. Since (propositional) modal logic can be embedded into FO^2, the decidability of FO^2 provides some understanding of the tractability of (propositional) modal logics. However, while several extensions of modal logic, like computational tree logic or CTL, remain decidable (for validity), corresponding extensions of FO^2 lead to undecidability. The guarded fragment of first-order logic is another approach to capture the nice properties of modal logics. It refers to the class of formulae in which all quantifiers are appropriately relativized by atoms. This fragment was later generalized to the loosely guarded fragment LGF [4], where all quantifiers are appropriately relativized by conjunctions of atoms. Both fragments are decidable and enjoy several useful syntactic and model theoretic properties that do not, in general, hold for FO^2. However, already very modest extensions of GF lead to undecidability [2]: GF with three

variables and transitive relations, and GF with three variables and counting quantifiers, are both undecidable extensions of GF.

The two-variable guarded fragment GF^2 is, however, powerful enough to encode the Kripke semantics of propositional multi-modal logics. For multi-modal logics with modalities of type K4, S4, and S5, GF^2_- with transitive relations appears as a natural choice for a representation language. We have shown that GF^2_- with transitive relations is undecidable [3]. Moreover, this is the case even when all non-unary relations are transitive binary relations. Hence this class is too large to capture these multi-modal logics adequately. On the other hand, when encoding propositional modal logics, the non-unary relations only appear as guards, such guarded formulas are said to be monadic. The second main result in [3] is that monadic GF^2 with binary relations that are transitive, symmetric and/or reflexive, is decidable. The latter result has been proved by an encoding of this class in *SkS* (similar to how this can be done for CTL). We have also shown that LGF without equality becomes undecidable as soon as a single relation is allowed to be transitive.

References

[1] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. ILLC Research Report ML-1996-03, University of Amsterdam, February 1996.

[2] E. Grädel. On the restraining power of guards. To appear in *Journal of Symbolic Logic*, July 1998.

- [3] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999. To appear.

[4] J. van Benthem. Dynamic bits and pieces. ILLC Research Report LP-1997-01, University of Amsterdam, January 1997.

[5] M. Y. Vardi. Why is modal logic so robustly decidable? In N. Immerman and P. G. Kolaitis, editors, *Descriptive Complexity and Finite Models : Proceedings of a DIMACS Workshop, January 14-17, 1996, Princeton University*, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 149–183. American Mathematical Society, Providence, Rhode Island, 1997.

7.6 Hybrid Logics

Investigator: Miroslava Tzakova

The expressivity of the basic modal language is comparatively limited, and must be typically boosted by the addition of various (application dependent) new modalities, such as the *Until* operator. Hybrid languages were proposed in [7] as an alternative way of increasing the expressivity of modal systems. The mechanism is to add labels for states in Kripke models and to view these labels as formulae. For example, given a label x – a special sort of formula – the formula $x \rightarrow \neg\Diamond x$ is well-formed. The latter formula expresses irreflexivity, which is a first-order property that is not modally definable. Hybrid languages combine features of both modal and classical logic: they possess Kripke semantics and in addition contain quantifiers to bind the labels. We consider quantifiers over arbitrary states in Kripke models as well as quantifiers that bind locally, namely, that bind labels to the current state (of evaluation) or to accessible states.

As a consequence, hybrid logics greatly increase the expressivity of modal systems, for example, they can define counting modalities, such as 'there are at least two successors', and the *Until* operator. For if $\exists x$ is an existential quantifier over all states in a Kripke model, then *Until* can be defined by the formula:

$$\mathit{Until}\ (\varphi, \psi) \ := \ \exists x(\Diamond(x \wedge \varphi) \wedge \Box(\Diamond x \rightarrow \psi)).$$

This definition says: it is possible to bind the label x to a state in such a way that (1) the state labeled x is a successor that satisfies φ, and (2) ψ holds at all states preceding this labeled state.

In fact, hybrid logics can be viewed as fragments of classical logic ranging from logics that are proper fragments to systems having full first-order expressive power. Moreover, hybrid languages not only can express the *Until* operator, and therefore possess the expressivity of linear time temporal logic, but by enriching them with labels for paths and quantifiers over paths, they can capture the expressivity of branching time temporal logic. For more detailed discussions on the relevance of hybrid languages to linear and branching time temporal logic as well as to knowledge representation we refer to [1, 3, 4, 5].

Our work aims at answering some fundamental questions concerning hybrid languages. In [2, 4, 5] we present Hilbert-style axiomatizations for hybrid logics of various quantifiers and prove them complete. Moreover, we investigate tableau proof systems for hybrid logics. In [6] we discuss tableau calculi for both hybrid logics that are proper fragments of classical logic as well as for systems having full first-order expressive power. We prove completeness of the proposed calculi and thus show that hybrid formalisms behave also proof-theoretically well. For some decidable logics we give tableau-based decision procedures.

References

[1] V. Goranko. An interpretation of computational tree logics into temporal logics with reference pointers. Technical Report 2/96, Verslagreeks van die Department Wiskunde, RAU, Department of Mathematics, Rand Afrikaans University, Johannesburg, South Africa, 1996.

- [2] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6(4):625–650, 1998. Revised Version of MPI-I-97-2-007.
- [3] P. Blackburn and M. Tzakova. A hybrid concept language. In E. Boros and R. Greiner, editors, *Proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida, 1998. `http://rutcor.rutgers.edu/~amai`.
- [4] P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):23–49, 1998.
- [5] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999. Revised Version of MPI-I-98-2-006.
- [6] M. Tzakova. Tableau calculi for hybrid logics. In N. V. Murray, editor, *Proceedings of the International Conference TABLEAUX'99 - Analytic Tableaux and Related Methods*, LNAI, Saratoga Springs, NY USA, 1999. Springer. To appear.

[7] A. Prior. *Past, present and future.* Oxford University Press, Oxford, 1967.

8 Logic and Uncertainty

We investigate formal frameworks for reasoning with uncertain information. Such formal frameworks can be purely qualitative non-classical logics that abstract from any numerical description of likelihood and utility, and thus try to capture the essential logical structure of reasoning under uncertainty. They can also assume more quantitative forms, and ultimately lead to data structures and algorithms for computing with concrete numerical values.

Our main motivation for studying such frameworks comes from artificial intelligence, where they are used to model human reasoning and decision making under uncertainty. The more numerical frameworks we have studied (notably Bayesian networks), however, are closely related to formal methods that are also used in areas as diverse as computational economics, computational biology, coding theory, and probabilistic verification.

In particular, our work has centered on three topics: deontic logic as a basis for qualitative decision theory, probabilistic foundations for default reasoning, and formal systems for (quantitative) probabilistic reasoning. Each of these topics is described in more detail below.

Richard Booth, who has recently joined our group, has previously worked on the "ent model" of belief – a numerical but non-probabilistic model of human belief introduced by J. Paris and A. Vencovská.

8.1 Desires and obligations

Investigators: Leon van der Torre, Emil Weydert

In the PhD thesis [16] deontic logic has been studied, a modal logic in which the modal sentence Op is read as 'it is obligatory that p is the case.' It has been developed as a branch of philosophical logic, and it has recently been studied in computer science. Topics identified are legal knowledge-based systems, the specification of fault tolerant systems, the specification of security policies, the automatization of contracting and the specification of normative integrity constraints for databases [2]. However, deontic logic is not sufficient for all applications that are based on normative reasoning. The problem is that deontic logic only formalizes reasoning about obligations, that is, which obligations follow from a set of obligations. However, there is a demand to formalize reasoning with obligations. For example, a legal expert system may face the diagnostic problem to determine whether a suspect has violated a legal rule, and a robot may have to solve the planning problem how to fulfill the desires of his owner. This raised interest in the area of Qualitative Decision Theory (QDT). Decision theory and related theories in economics concentrate on a notion of expected utility that is representable using quantitative preferences and probabilities. More recent traditions in Artificial Intelligence have explored qualitative decision methods including control rules, rule orderings, default preferences, and qualitative approaches to probability. The logics of desires and obligations have been studied from a proof-theoretic perspective (variants of conditional logic and labelled deductive systems), as well as from a semantic perspective (preference-based and decision-theoretic – utilitarian and probabilistic – semantics).

First, we studied the relation between qualitative decision theory and deontic logic. In [1] we argue that conditional desires as well as obligations (and goals, ideals, preferences, actions, beliefs, ...) can be formalized as inference-like processes on propositions, in which inputs are not in general included among outputs, but in which outputs may be recycled as inputs. We outline a general theory of such processes, both unrestricted and subject to consistency constraints. This proof-theoretic perspective builds on labelled logics for desires and obligations [4, 5, 6]. A first comparison between the semantic (preference-based) perspective of these two areas has been given

in [7, 13] (see [8] for the relation with defeasible reasoning).

Secondly, we studied desires in Lang's framework for qualitative decision theory. In qualitative decision-theoretic planning desires – qualitative abstractions of utility functions – are combined with defaults – qualitative abstractions of probability distributions – to calculate the expected utilities of actions. In [15, 18] we consider Lang's framework of qualitative decision theory, in which utility functions are constructed from desires. Unfortunately there is no consensus about the desired logical properties of desires, in contrast to the case for defaults. To do justice to the wide variety of desires we define parameterized desires in an extension of Lang's framework. There are three parameters. The strength parameter encodes the importance of the desire, the lifting parameter encodes how to determine the utility of a set from the utilities of its elements, and the polarity parameter encodes the relation between gain of utility for rewards and loss of utility for violations. The parameters influence how desires interact, and they thus increase the control on the construction process of utility functions from desires.

Third, we studied obligations in deontic logic. The main problems to apply deontic logic are the contrary-to-duty and dilemma paradoxes. The conceptual issue of the contrary-to-duty paradoxes is how to proceed once a norm has been violated. Clearly this issue is of great practical relevance, because in most applications norms are violated frequently. Usually it is stipulated in the fine print of a contract what has to be done if a term in the contract is violated. Usually the contracting parties do not want to consider a violation as a breach of contract, but simply as a disruption in the execution of the contract that has to be repaired. The conceptual problem of the dilemma paradoxes is to determine the coherence conditions of a normative system. For example, when drafting regulations a coherence check indicates whether they have this desired property, or whether they should be further modified. The recently developed preference-based deontic logics [9, 17, 10, 12], and in particular the prescriptive obligations in update semantics [3, 11, 14] have delivered some promising approaches for these long-standing problems in normative reasoning and their notorious deontic paradoxes, and a further increase in the above mentioned applications may be expected.

References

[1] D. Makinson and L. W. N. van der Torre. The logic of reusable propositional output. Draft., January 1999.

- [2] B. S. Firozabadhi and L. W. N. van der Torre. Towards a formal analysis of control systems. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 317–318, Brighton, England, 1998. Wiley.
- [3] L. van der Torre and Y.-H. Tan. An update semantics for deontic reasoning. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, volume 49 of *Frontiers in Artificial Intelligence and Applications*, pages 73–90. IOS, Amsterdam, 1999.
- [4] L. W. N. van der Torre. Labeled logics of conditional goals. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 368–369, Brighton, England, 1998. Wiley.
- [5] L. W. N. van der Torre. Phased labeled logics of conditional goals. In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Proceedings of the 6th European Workshop on Logics in AI: Logics in Artificial Intelligence (JELIA-98)*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 92–106, Schloss Dagstuhl, 1998. Springer.
- [6] L. W. N. van der Torre. Defeasible goals. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Proceedings of the ECSQARU'99*, LNAI, London, 1999. Springer. To appear.

- [7] L. W. N. van der Torre and Y.-H. Tan. Distinguishing different roles in normative reasoning. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL-97)*, pages 225–232, Melbourne, Australia, 1997. The international association for artificial intelligence and law; the university of Melbourne law school, ACM Press.
- [8] L. W. N. van der Torre and Y.-H. Tan. Reasoning about exceptions. In G. Brewka, C. Habel, and B. Nebel, editors, *Proceedings of the 21st Annual German Conference on Artificial Intelligence (KI-97): Advances in Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 405–408, Freiburg, September 1997. Springer.
- [9] L. W. N. van der Torre and Y.-H. Tan. Prohairetic deontic logic (pdl). In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Proceedings of the 6th European Workshop on Logics in AI: Logics in Artificial Intelligence (JELIA-98)*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 77–91, Schloss Dagstuhl, 1998. Springer.
- [10] L. W. N. van der Torre and Y.-H. Tan. The temporal analysis of Chisholm's paradox. In *Proceedings of the 15th National Conference, and 10th Conference on Innovative Applications of Artificial Intelligence (AAAI-98) and (IAAI-98)*, pages 650–655, Madison, Wisconson, 1998. American Association for Artificial Intelligence (AAAI), AAAI Press/ MIT Press.
- [11] L. W. N. van der Torre and Y.-H. Tan. An update semantics for prima facie obligations. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 38–42, Brighton, England, 1998. Wiley.
- [12] L. W. N. van der Torre and Y.-H. Tan. Contextual deontic logic: violation contexts and factual defeasibility. In M. Cavalcanti, editor, *Formal Aspects in Context*, Applied Logic Series. Kluwer, Dordrecht, 1999.
- [13] L. W. N. van der Torre and Y.-H. Tan. Diagnosis and decision making in normative reasoning. *Journal of artificial intelligence and law*, 1999. To appear.
- [14] L. W. N. van der Torre and Y.-H. Tan. Rights, duties and commitments between agents. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, 1999. IJCAII and the Scandinavian AI Societies, Morgan Kaufmann. To appear.
- [15] E. Weydert and L. W. N. van der Torre. Goals, desires, utilities and preferences. In J. Lang, editor, *Proceedings of the ECAI-98 Workshop : Decision theory meets artificial intelligence - qualitative and quantitative approaches*, pages 57–64, Brighton, United Kingdom, 1998. ECAI.

[16] L. W. N. van der Torre. *Reasoning about obligations; defeasibility in preference-based deontic logic.* PhD thesis, Erasmus University Rotterdam, 1997.

[17] L. W. N. van der Torre and Y.-H. Tan. Contrary-to-duty reasoning with preference-based dyadic obligations. Draft., January 1999.

[18] L. W. N. van der Torre and E. Weydert. Risk parameters for utilitarian desires in a qualitative decision theory. Draft., January 1999.

8.2 Probabilistic default reasoning

Investigator: Emil Weydert

Probabilistic approaches to default reasoning interpret default implications, e.g. $\mathit{Bird}(a) \Rightarrow \mathit{Canfly}(a)$ (Bird(a) normally implies Canfly(a)), as constraints over quasi-probabilistic plausibility measures, e.g. $QP(\mathit{Bird}(a) \wedge \neg \mathit{Canfly}(a)) < QP(\mathit{Bird}(a) \wedge \mathit{Canfly}(a))$. Here we may distinguish between the coarse-grained ranking measure semantics, which uses probabilistic order-of-magnitude valuations, and the fine-grained nonstandard probability semantics, which exploits extended probability distributions with infinitesimal values. Nonstandard probability measures are linked to ranking measures by a projection function. The idea is then to determine the most reasonable or

plausible models of the given constraints within the chosen quasi-probabilistic framework, which defines a nonmonotonic plausible inference relation.

In previous work, we have proposed preference strategies for the coarse-grained and the fine-grained account. Notably JJ-constructibility [7, 2] - only consider those ranking measure models accessible by a minimal, irredundant iterated update process from the uniform, i.e. maximally ignorant valuation - and soft entropy maximization [6] - a more robust variant of entropy maximization available within the nonstandard context (necessary to grasp the uncertain character of default knowledge and to avoid spurious conclusions). During the past two years, we have addressed three major questions, which have been investigated in the context of the DFG-project "Defaults and Probability". Namely, what is the exact relation between these different approaches? Is it possible to single out a canonical, "most plausible" model of a given default knowledge base? How may we extend the basic techniques to more expressive languages, i.e. to boolean or first-order constraints?

Concerning the first question, among others we were able to show that for finite sets of ranking constraints of the form $r(A) + a \leq r(A')$ with $a \neq 0$, the JJ-constructible ranking measure models correspond to the projections of the soft-entropy-maximal nonstandard probability models [3, 8]. This result is rather surprising given the distinct intuitive backgrounds. But it is very useful because JJ-constructibility is much easier to handle than entropy maximization for a continuum of parameter combinations. Whether there is a similar result for constraints of the type $r(A) \leq r(A')$ remains an open problem.

In standard probabilistic reasoning, if we want to select a single most plausible element from a closed convex set of probability distributions, possessing no additional information, axiomatic characterization results suggest that the maximum entropy model may constitute the most reasonable choice. But how should we proceed in the ranking measure context? Because the usual probabilistic semantics for defaults produces closed convex constraint sets, the most straightforward strategy would be to maximize entropy in the nonstandard context and to consider the corresponding ranking measure projection. Unfortunately, this solution is not only difficult to compute, but it is also very sensitive to small changes of the problem description.

We have therefore defined a specific, maximally uniform JJ-construction process, which builds the canonical JZ-model [4] for any consistent finite default knowledge base. This approach is not representation dependent and determines a very powerful default entailment notion. It combines the transparency of rational closure, a popular account proposed by Daniel Lehmann [1], with the ability to handle inheritance to exceptional subclasses in a suitable way. We also have generalized this algorithm to deal with constraints of the form $r(A) \leq r(A')$. They require a more sophisticated prioritized construction procedure to avoid that weak inequalities cause loops. Based on this, we have been able to extend Spohn's iterated belief revision procedure - for ranking measures representing epistemic states - to deal with multiple evidence expressed by sets of conditional ranking constraints [5].

References

[1] D. Lehmann. What does a conditional knowledge base entail ? In H. J. L. Ronald J. Brachman and R. Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 212–222, Toronto, Canada, 1989. Morgan Kaufmann.

- [2] E. Weydert. Qualitative entropy maximization - A preliminary report. In E. Weydert, G. Brewka, and C. Witteveen, editors, *Proceedings of the Third Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR-97)*, pages 63–72, Saarbrücken, Germany, 1997. Max-Planck-Institut für Informatik.

- [3] E. Weydert. Minimal information entailment : A preliminary account. In J. Delgrande and M. Truszczynski, editors, *Proceedings of the 7th International Workshop on Nonmonotonic Reasoning (Workshop on Formal Aspects and Applications of Nonmonotonic Reasoning)*, pages 64–72, Trento, Italy, 1998.
- [4] E. Weydert. System JZ : How to build a canonical ranking model of a default knowledge base. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pages 190–201, Trento, Italy, 1998. Morgan Kaufmann.
- [5] E. Weydert. JZBR - Iterated belief change for conditional ranking constraints. In H. Rott, C. Albert, G. Brewka, and C. Witteveen, editors, *Proceedings of the Fourth Dutch-German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR 99)*, pages 57–66, Amsterdam, The Netherlands, 1999. ILLC Scientific Publications.

[6] E. Weydert. General belief measures. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 575–582, University of Washington, Seattle, WA, 1994. Morgan Kaufmann.

[7] E. Weydert. System j - revision entailment: Default reasoning through ranking measure updates. In D. Gabbay and H. J. Ohlbach, editors, *Practical Reasoning - International Conference on Formal and Applied Practical Reasoning, FAPR'96*, volume 1085 of *Lecture Notes in Computer Science*, pages 637–649, Bonn, Germany, June 1996. Springer.

[8] E. Weydert. Robust default entailment - Rank-based minimal information approaches. January 1999.

8.3 Formal Probabilistic Reasoning

Investigator: Manfred Jaeger

Our work in this area can be roughly divided into two distinct parts: foundational issues in probabilistic reasoning, and the development of a representation and inference system for random relations.

Foundations Main focus of our work here has been the problem of measure selection. By measure selection we mean the form of probabilistic inference where from partial information about a probability distribution (e.g. a set of linear constraints) we do not merely infer those probability bounds for events of interest that are implied through the rules of probability theory, but aim to tighten these bounds by basing our inference only on some preferred elements in the set of all probability measures satisfying the constraints.

As a formal rule for such a selection process most often entropy maximization is proposed. As an intuitive justification for this selection principle it is often stated that it implements a minimal information gain principle, i.e. by selecting the maximum entropy distribution only a minimal amount of additional information will be assumed.

In [5] a quite different principle is proposed on which to base the selection rule: it is argued that this selection process is essentially a statistical parameter estimation problem, where from some given data (the constraints) we wish to determine the parameter that determines the distribution of the data (the preferred or "true" distribution). Thus, this perspective on the measure selection problem takes into account that the "true" probability distribution will also affect the likelihood of obtaining certain constraints. While it is almost immediate that entropy maximization is incompatible with this perspective, it is not easy to find general selection rules that will implement this statistical perspective without making too specific statistical assumptions. The investigation of such rules is the subject of ongoing work.

Another basic form of probabilistic inference – conditioning – has been investigated in joint work with Ian Pratt [2]. Here we have shown that only under very restrictive assumptions will

conditioning, as it is widely employed in probabilistic expert systems, yield probabilities that are correct in the sense that the same numbers would also be obtained if we worked in a probability space in which our evidential situation can be modeled completely.

Relational Bayesian Networks Bayesian networks are the most successful formal framework for representing and manipulating probability distributions on large attribute spaces. The expressiveness of standard Bayesian network, however, is limited to describing random attributes of individual objects. Random relations between several objects can only be described if a fixed, finite, domain of individual objects is assumed – in which case a random relation between objects becomes just a random attribute of the domain. However, very often, we want to deal with probabilistic models of relations in a general way, without fixing a single domain. This has previously led to proposals for knowledge based model construction (e.g. [7]). These are approaches in which probabilistic information is expressed in representation languages whose key elements are probabilistic Horn clauses. This rule-based representation paradigm tends to suffer from at least one of two problems: they often are not very expressive in that more complicated interactions between random relations can not be represented, and they often do not possess a very transparent semantics.

Relational Bayesian networks were proposed in [3] as a framework that combines great expressiveness with a very clean semantics. The basic idea is to represent the probabilistic dependencies of one random relation `r` on other relations `s,t, ...` by a single functional expression that determines the probability of each ground `r`-atom as a function of the interpretations of the relations `s,t, ...` .

The simplicity both of the syntax rules for the formation of these functional expressions, and of the semantics of the resulting relational Bayesian networks, makes this approach more amenable to investigations of its theoretical properties than the less manageable rule-based frameworks. In particular, it has been possible to show that the semantics of relational Bayesian networks can be extended to countably infinite domains, and a method has been found to effectively compute the probabilities defined in this case [6]. Another issue of great theoretical interest is the dependency of the probability of ground formulas on the size of the underlying domain. Here we have shown that for a certain subclass of relational Bayesian networks these probabilities will converge with increasing domainsize [4]. This result can also be read as a new kind of convergence theorem in finite model theory.

A general result on the complexity of probabilistic inference about random relations has also been derived [1]. It is shown that when we consider the complexity in terms of the size of the underlying domain, and assuming NETIME≠ETIME, then for representation formalisms with the expressiveness of relational Bayesian networks there do not exist inference methods with a better worst-case behavior than the commonly employed technique of constructing an auxiliary standard Bayesian network over ground atoms, and answering the query by applying standard algorithms to this network.

References

[1] M. Jaeger. On the complexity of probabilistic inference from knowledge bases. Draft, 1999.

[2] M. Jaeger and I. Pratt. Conditions for conditioning. Draft, 1999.

- [3] M. Jaeger. Relational bayesian networks. In D. Geiger and P. P. Shenoy, editors, *Proceedings of the 13th Conference of Uncertainty in Artificial Intelligence (UAI-13)*, pages 266–273, Providence, USA, 1997. Morgan Kaufmann.

- [4] M. Jaeger. Convergence results for relational Bayesian networks. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 44–55, Indianapolis, USA, 1998. IEEE Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.
- [5] M. Jaeger. Measure selection: Notions of rationality and representation independence. In G. S. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 274–281, Madison, USA, 1998. Morgan Kaufmann.
- [6] M. Jaeger. Reasoning about infinite random structures with relational bayesian networks. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pages 570–581, Trento, Italy, 1998. Morgan Kaufmann.

[7] L. Ngo and P. Haddawy. Probabilistic logic programming and bayesian networks. In *Algorithms, Concurrency and Knowledge (Proceedings ACSC95)*, Springer Lecture Notes in Computer Science 1023, pages 286–300, 1995.

9 Other Work

9.1 Logical Frameworks

Investigators: Sean Matthews, Luca Viganò

Research in 'Logical frameworks', which in our group is taken to be loosely defined as the theory of (interactive) theorem proving systems, has continued to be fruitful, and we have been able to develop considerably work that was only beginning in the last report.

First, is the theory of labelled deduction systems. These systems can be described as hybrid deduction systems which attempt, on the basis of a (generalized) Kripke semantics, to provide 'natural' deduction systems for families of non-classical logics, such as modal or substructural (e.g. relevance and linear) logics, where 'natural' means uniform, modular, and with the sort of properties which would allow them to be implemented directly in a generic proof development system such as the Edinburgh LF, which assumes a 'mathematical' deduction system. In a series of papers over the last two years [3, 6, 8, 5, 4, 7, 2], and in Viganò's PhD thesis [13], we have been able to present a systematic investigation of the basic proof theoretic properties of such logics, starting from propositional modal logics and progressively extending our framework to deal with quantification and generalized non-classical modalities such as relevant implication and negation. What is probably most interesting about our systems is the fact that they are not universal: we can distinguish classes of labelled deduction system which are complete only for subsets of logics with first-order presentations (in contrast to the standard treatment of 'semantic embedding'). We were able not only to document the exact nature of this restriction, and the way it affects the class of definable logics, but also to exploit it for other purposes. Specifically, we have used it as the foundation of a new proof theoretic method for establishing space complexity bounds for the decision problem in non-classical logics. We have applied our method to various modal logics, and been able to provide results that are comparable with the best known. Preliminary results of this work have been published in [5, 13, 2] and we have submitted a more systematic journal paper [1].

A second area of work that has been developed further is into the use of a theory of inductive definitions as a general framework for defining logics. An obvious problem with such theories is that, even though they offer more general metatheoretic facilities, they seem to lack the basic facilities needed in order to do simple object theory. We have shown that it is possible to reconstruct many of the facilities of type-theoretic frameworks by a combination of abstract theorems, and some metatheory. This work is described in [10]. This work is closely related to another question we have investigated, that of how to reconcile theory structuring in a theorem proving system with general (i.e. inductive) metatheorms for particular theories. We develop mechanisms for doing this, and investigate some of the payoffs in the papers [12, 11]. The second of these papers in particular has attracted interest, David Basin being invited to SRI International, Menlo Park, last summer in part to discuss it.

A third area of research has been on extensions to the standard type-theoretical logical framework theories. The problem is that while these notations are suitable for the standard logics of mathematics, they are less and less suitable for the logics of computer science, artificial intelligence, or philosophy. The problem though is to find reasonable, well founded extensions that fit the 'spirit' of the original idea. We have developed as an example, an extension of the standard logical framework that provides the machinery to formalize 'validity' (theoremhood) judgments, using general ideas from proof theory [9].

References

[1] D. A. Basin, S. Matthews, and L. Viganò. Complexity bounds for propositional modal logics: A proof-theoretic method. Journal version, submitted, 1999.

[2] D. A. Basin, S. Matthews, and L. Viganò. Modal logics k, t, k4, s4: labelled proof systems and new complexity results. One page abstract to appear in the proceedings of Logic Colloquium'98, published in the Bulletin of Symbolic Logic, 1999.

- [3] D. A. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.
- [4] D. A. Basin, S. Matthews, and L. Viganò. Labelled quantified modal logics. In G. Brewka, C. Habel, and B. Nebel, editors, *Proceedings of the 21st Annual German Conference on Artificial Intelligence (KI-97): Advances in Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 171–182, Freiburg, Germany, 1997. Springer.
- [5] D. A. Basin, S. Matthews, and L. Viganò. A new method for bounding the complexity of modal logics. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory (KGC-97)*, volume 1289 of *Lecture Notes in Computer Science*, pages 89–102, Vienna, Austria, 1997. Springer.
- [6] D. A. Basin, S. Matthews, and L. Viganò. Labelled modal logics: quantifiers. *Journal of Logic, Language and Information*, 7:237–263, 1998.
- [7] D. A. Basin, S. Matthews, and L. Viganò. A modular presentation of modal logics in a logical framework. In J. Ginzburg, Z. Khasidashvili, C. Vogel, J.-J. Levy, and E. Vallduvi, editors, *Proceedings of the 1st Tbilisi Symposium on Language, Logic and Computation: Selected Papers*, Studies in Logic, Langugage and Information, pages 293–307, Tbilisi, Georgia, 1998. folli, CSLI.
- [8] D. A. Basin, S. Matthews, and L. Viganò. Natural deduction for non-classical logics. *Studia Logica*, 60(1):119–160, 1998.
- [9] S. Matthews. Extending a logical framework with a modal connective for validity. In M. Abadi and T. Ito, editors, *Proceedings of the 3rd Symposium on Theoretical Aspects of Computer Software (TACS-97)*, volume 1281 of *Lecture Notes in Computer Science*, pages 491–514, Sendai, Japan, September 1997. Tohoku University, ACM SIGACT, the Information Processing Society of Japan, the Japan Society of Software Science, Association for Symbolic Logic, Springer.
- [10] S. Matthews. A practical implementation of simple consequence relations using inductive definitions. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (CADE-14)*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 306–320, Townsville, Australia, July 1997. Association for Automated Reasoning, Springer.
- [11] S. Matthews and D. A. Basin. Scoped metatheorems. *Electronic Notes in Computer Science*, 15:1–14, 1998.
- [12] S. Matthews and D. A. Basin. Structuring metatheory on inductive definitions. *Information and Computation*, 1999. To appear.
- [13] L. Viganò. *A Framework for Non-Classical Logics.* PhD thesis, Universität des Saarlandes, 1997.

9.2 Complexity of Nonrecursive Logic Queries

Investigators: Sergei Vorobyov, Andrei Voronkov

A large number of complexity results have been established for logic query languages. New relational query languages, for example SQL-3, extend traditional languages in several directions. One of them is the introduction of complex values, like sets. In [2, 1] we investigated complexity of the SUCCESS problem for logic query languages with complex values: check whether a query

defines a nonempty set. The SUCCESS problem for recursive query languages with complex values is undecidable, in general, so we study the complexity of nonrecursive queries. By complex values we understand values such as trees, finite sets, and multisets. Due to the well-known correspondence between relational query languages and datalog, our results can be considered as results about relational query languages with complex values. The papers [2, 1] give a complete complexity classification of the SUCCESS problem for nonrecursive logic programs over trees depending on the underlying signature, presence of negation, and range restrictedness. We also proved several results about finite sets and multisets.

References

- [1] S. Vorobyov and A. Voronkov. Complexity of nonrecursive logic programs with complex values. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-98)*, pages 244–253, Seattle, Washington, U.S.A., June 1998. Assoc. Comput. Machinery, Boeing, ACM Press.

[2] S. Vorobyov and A. Voronkov. Complexity of nonrecursive logic programs with complex values. Research Report MPI-I-97-2-010, Max-Planck Institut für Informatik, Im Stadtwald, Saarbrücken, D-66123, Germany, November 1997.

9.3 Natural Nonelementary Theories

Investigator: Sergei Vorobyov

What is the highest possible lower complexity bound for a 'natural' decidable theory? Until recently, it was widely believed that theories like Büchi's and Rabin's monadic second-order arithmetics are the most complicated such theories, with lower bounds being stacks of twos growing linearly with the length of a formula. In [5] we showed that a decidable rudimentary theory Ω of finite typed sets [2, 4, 6, 3] requires space exceeding infinitely often (lower bound)

$$\exp_\infty(\exp(cn)) = 2^{\left.2^{\cdot^{\cdot^{2}}}\right\}\mathit{height}\ 2^{cn}} \quad \text{for some constant } c > 0.$$

This gives the highest currently known lower bound for a decidable logical theory and affirmatively answers to [1, Problem 10.13, p. 75]:

> Is there a 'natural' decidable theory with a lower bound of the form $\exp_\infty(f(n))$, where f is not linearly bounded?

The highest previously known lower (and upper) bounds for 'natural' decidable theories, like *WS1S*, *S2S*, have form $\exp_\infty(dn)$, with just linearly growing stacks of twos.

Originally, in [5], this lower bound for Ω was settled by using the powerful uniform lower bounds method due to [1], and probably would not have been discovered otherwise. Although very concise, the original proof left a possibility that the method was pushed beyond the limits it was originally designed and intended for, and some hidden assumptions were violated. In [7] we gave an independent direct proof by generic reduction of the same lower bound. This alternative proof also helped to figure out several gaps and hidden assumptions overlooked in [5].

The lower bound for Ω was used in [5] to settle tight lower bounds for the $\beta(\eta)$-equality in the simply typed lambda calculus, and also to settle a strong (nonelementary) lower bound for the currently open higher-order matching problem due to G. Huet, see also [8].

References

[1] K. J. Compton and C. W. Henson. A uniform method for proving lower bounds on the computational complexity of logical theories. *Annals of Pure and Applied Logic*, 48:1–79, 1990.

[2] L. Henkin. A theory of propositional types. *Fundamenta Mathematicae*, 52:323–344, 1963.

[3] H. Mairson. A simple proof of a theorem of Statman. *Theoretical Comput. Sci.*, 103:387–394, 1992.

[4] A. R. Meyer. The inherent computational complexity of theories of ordered sets. In *International Congress of Mathematicians*, pages 477–482, Vancouver, 1974.

[5] S. Vorobyov. The 'hardest natural decidable theory. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97)*, pages 294–305, Warsaw, Poland, June-July 1997. IEEE Comput. Soc. Press.

[6] R. Statman. The typed λ-calculus is not elementary recursive. *Theoretical Comput. Sci.*, 9:73–81, 1979.

[7] S. Vorobyov. The most nonelementary theory (a direct lower bound proof). Research Report MPI-I-98-2-007, Max-Planck Institut für Informatik, April 1998.

[8] S. Vorobyov. The 'hardest' natural decidable theory and improved lower bounds in λ-calculus. Extended and revised version of the LICS'97 paper, submitted, 1999.

9.4 Formal Foundations for the State as Algebra Approach

Investigator: Hubert Baumeister

One way to model the state space of a dynamic system is as an abstract datatype. Each model of the abstract datatype is an admissible state of the dynamic system. The theory of abstract datatypes is well developed, has a rigorous formal basis and is institution independent, i.e. independent of the particular logic used for defining abstract datatypes, like, e.g., many sorted equational logic, or order sorted first order logic with partial functions. The use of algebras to model the state of a dynamic system is quite common and is referred to as the state as algebra approach.

In [1] we have given institution independent formal foundations for the state as algebra approach. We have defined, based on a given institution describing the state space of a dynamic system, a new institution, where abstract datatypes are interpreted as relations on models of the abstract datatypes from the base institution. The advantage of this approach is that the results from the institution independent part of the theory of abstract datatypes can be reused. For example, we have defined a specification language for the specification of relations, which is based on the usual operations defined on abstract datatypes, like, union, signature extension, hiding etc. Further, we have shown how to apply an institution independent proof calculus for proving properties of abstract datatypes to prove properties of relations and entailment of relations.

We have defined an institution for the logical system of the model-oriented specification language Z, which allowed us to formally relate the state as algebra approach to the model-oriented way of specifying dynamic systems used by Z. This has resulted in a specification method which is in many aspects similar to Z's method, but can be used with any suitable logical system, including the logical systems used for the state as algebra approach, as well as the logical system used by Z.

References

[1] H. Baumeister. Relations between abstract datatypes modeled as abstract datatypes, 1998. PhD thesis, submitted.

9.5 Linear Logic

Investigator: Giorgio Delzanno

Linear logic is a powerful and expressive logic connected to a variety of topics in computer science. From a proof-theoretical point of view, LL derives from classical logic by eliminating the structural rules of weakening and contraction. As a consequence, in the resulting proof-calculi it is possible to treat formulae as resources. Contraction and weakening are re-introduced in a restricted way, i.e., they can be applied only to the subclass of formulae prefixed by special modalities.

We aim at studying linear logic as a specification language for advanced concepts of programming (e.g., concurrency and object-orientation). Our approach is based on a refinement of linear logic sequent-calculi based on the proof-theoretic characterization of logic programming. Specifically, we consider uniform proof systems in the style of extensions of logic programming based on intuitionistic logic. Given a sequent $s = \Gamma \rightarrow G$, a uniform proof for s is a cut-free proof built according to the following strategy: (1) decompose the 'goal' G to atomic formulas; (2) apply the formulas in Γ (e.g. using resolution-like steps) and go back to (1), until an axiom is reached.

A well-founded combination of higher-order logic programming and linear logic is used to give accurate encodings of different calculi for object-oriented and agent-oriented programming. In [3], we have isolated a fragment of higher-order linear logic that serves as specification language for a wide class of object-oriented primitives and constructs. Then we have introduced an object-based language (in Abadi-Cardelli style) and have shown that the fragment taken into consideration is powerful enough to encode this language. In [5, 1, 2], we have extended the previous idea to concurrent object- and agent-systems. In [4], we analyze the relation with other formalisms used to express computational aspects (term rewriting).

References

- [1] M. Bozzano, G. Delzanno, M. Martelli, V. Mascardi, and F. Zini. Multi-Agent Systems Development as a Software Engineering Enterprise. In G. Gupta, editor, *Proceedings of the First International Workshop on Practical Aspects of Declarative Languages (PADL 99)*, volume 1551 of *Lecture Notes in Computer Science*, pages 46–60, San Antonio, Texas, USA, October 1998. Compulog Americas and Association for Logic Programming (ALP), Springer.
- [2] M. Bozzano, G. Delzanno, M. Martelli, V. Mascardi, and F. Zini. Logic Programming and Multi-Agent Systems: a Synergic Combination for Applications and Semantics. In *The Logic Programming Paradigm: a 25-Year Perspectiv*, Lecture Notes in Computer Science, chapter 1, pages 5–32. Springer, Berlin, 1999. To appear.
- [3] M. Bugliesi, G. Delzanno, L. Liquori, and M. Martelli. Object Calculi in Linear Logic. *Journal of Logic and Computation*, 1999. To appear.
- [4] G. Delzanno. Specification of Term Rewriting in Linear Logic. In D. Galmiche, editor, *Proceedings of Workshop on Proof-Search in Type-Theoretic Languages*, volume 17 of *Eletrical Notes in Theoretical Computer Science*, Lindau, Germany, 1998. Elsevier Science.
- [5] G. Delzanno, D. Galmiche, and M. Martelli. A Specification Logic for Concurrent Object-oriented Programming. *Mathematical Structures in Computer Science*, 1999. To appear.

10 Systems

Along with 'pure' research, the Logic of Programs group is developing various pieces of software in order to demonstrate the practical feasibility of our theoretical concepts.

In this section we describe our progress with regard to these systems, their availability, and (where applicable) their performance as measured against other competitive systems.

10.1 SPASS Version 1.0.0

Investigator: Christoph Weidenbach
Research assistants: Bijan Afshordel, Uwe Brahm, Christian Cohrs, Thorsten Engel, Enno Keen, Christian Theobalt, Dalibor Topić

SPASS is an automated theorem prover for full sorted first-order logic with equality that extends superposition by a sort concept and by a splitting rule for case analysis [7]. It is one of most powerful systems currently available. For example, it won four prices at recent CADE theorem prover system competitions, more than any other system. SPASS is meant to be useful for two groups of users. People using it as a tool and people using it as a development platform. Many of our recent efforts went into improving ease of use for both groups.

Compared to the version of SPASS described in the previous progress report a large amount of existing code was re-implemented or adjusted. In addition, we have added many new features, some of which are described below.

One of our long term goals is to provide as much of the functionality of SPASS as possible in form of a documented C-library. Towards this end, we improved modularization, debugging support and code documentation. Apart from the documentation contained in the code, there is currently not much extra documentation available. The memory management module is an exception in this regard. The module allocates memory in pages of uniform size that are then cut into appropriate pieces. In addition to a gain in performance, the module supports many debugging features that are otherwise only supported by commercial software. For example, it can detect writes over memory blocks or can point to memory leaks.

For SPASS we have adopted the GNU command line options package for C. So all SPASS options can now be given as command line options to SPASS. In addition to options selecting inference or reduction rules, the selection of various strategies (e.g. set of support), and various possibilities to influence the way SPASS's output looks like, there is the possibility to feed SPASS via pipes and to use the prover in an interactive way: first, a set of axioms is given to SPASS and then the prover can be subsequently given conjectures to prove with respect to such an axiom set. This is particularly useful in a context where SPASS is integrated as an inference engine for some other system.

We added an abstract ordering interface module to SPASS, where in particular the lifting of an arbitrary reduction ordering to literals/clauses is implemented. In addition to the Knuth-Bendix ordering (KBO) implemented in Version 0.55, we added an implementation of the recursive path ordering with status (RPOS).

SPASS now has the clause normal form translation built in, so it can be directly applied to first-order formulae, and it can be used to convert formulae into CNF. The current implementation contains all features described by Nonnengart, Rock and Weidenbach [5]: Optimized and strong Skolemization and the improved implementation of formula renaming, the replacement of subformulae by new predicate symbols (cf. section 3.1). Furthermore, we extended renaming such that it now first searches for generalizations of renaming candidates and then simultaneously replaces

all instances of the found generalization. For problems containing equality we added a number of simplification rules that eliminate occurrences of equations.

It is often useful to expand atom definitions before CNF transformations, and/or to apply atom definitions to conjecture formulae/clauses. An atom definition is meant to be formula of the form

$$\forall x_1, \ldots, x_n[\phi \supset (P([x_1, \ldots, x_n]) \equiv \psi)]$$

where $P([x_1, \ldots, x_n])$ denotes an arbitrary atom with predicate symbol P containing the variables $x_1, \ldots, x_n$. We require that P does not occur in ψ. SPASS searches an input file for formulae that can be transformed in the above form and then allows the user via options to replace occurrences of atoms $P([t_1, \ldots, t_n])$ by $\psi\sigma$ if in the replacement context the formula $\phi\sigma$ is valid where $\sigma = \{x_i \mapsto t_i \mid 1 \leq i \leq n\}$.

We added a variety of new inference rules to SPASS: Ordered/unordered hyper resolution, unit resolution, merging paramodulation, ordered/unordered paramodulation. All inference rules can be combined with selection strategies for negative literals. A further new inference rule is depth bounded unit resolution, a variant of unit resolution that requires the term depth of a unit resolvent to be less or equal to the maximal term depth of its parents. Although this rule is not complete, even in a Horn setting, it is guaranteed to terminate on any clause set and turned out to be very useful in practice for subproofs in the context of optimized Skolemization [5] and the applicability test for definitions (see above).

For many applications, like, e.g., automatic type inference (e.g., [1]) it is necessary/useful that the prover is a decision procedure for the input formula classes. If classes do not belong to a decidable fragment, safe approximations can be used to guarantee termination. We concentrated on monadic clause classes and implemented various methods to transform arbitrary clause sets into monadic clause sets. These can then be further approximated into decidable monadic clause sets. For example, a clause

$$\neg R(x, f(x, y)) \vee \neg S(x) \vee R(g(x), h(y))$$

can first be equivalently transformed into a monadic clause

$$\neg T(r(x, f(x, y))) \vee \neg S(x) \vee T(r(g(x), h(y)))$$

and then be approximated by the clauses

$$\neg T(r(x, f(x, y))) \vee \neg S(x) \vee \neg P(z) \vee \neg Q(v) \vee T(r(z, v))$$
$$\neg T(r(x, f(x, y))) \vee \neg S(x) \vee P(g(x))$$
$$\neg T(r(x, f(x, y))) \vee \neg S(x) \vee Q(h(y))$$

that then overestimate the relation R, represented by the function r in the approximation. The fragment formed by clauses of the final form is decidable and effective representations for R can be derived by a saturation of the clause set [6]. Approximation techniques are also a prerequisite for semantic approaches to guide the search for a prover itself [4] (cf. section 3.2).

We completely re-implemented the extraction of proofs from SPASS runs. The extracted proofs are now less redundant with respect to the application of the splitting rule, and our proof module is now able to deal with proofs of several hundred thousand steps in reasonable time. If splitting occurs in a proof, the proof has a tree (tableau) like structure. It is now possible to transform such proofs into trees that can then be graphically displayed.

Furthermore, we built an automated proof checker based on logical implication between clauses. SPASS proofs are translated into a sequence of proof obligations, one for each proof step. The tree structure caused by splitting applications is separately checked by an independent algorithm. Every single proof step obligation is then checked by a (different) prover. We usually employ Otter [3] for this task. The advantage of this method is that it is completely independent from variations of the used calculus and is able to check proofs up to several hundred thousand steps in reasonable time. Since the proof checker does not depend on SPASS nor on the calculus used in SPASS, any clause

based refutation proof that relies on logical implication can be checked using the checker.

We added a www interface to the SPASS homepage. The interface includes the possibility for a file upload and offers the full functionality of the prover including help files. In order to restrict the load of the server, SPASS runs are currently limited to 30 seconds.

We added a bunch of tools: FLOTTER is still our CNF-translator. Now it is simply implemented by a link to SPASS. `pcheck` is our proof checker (see above). `dfg2otter` transforms SPASS input files into Otter syntax. Our motivation for this was to employ Otter as a proof checker. `dfg2tptp` transforms SPASS input files into TPTP-Syntax. `prolog2dfg` transforms prolog programs into SPASS input files. This is currently restricted to purely logical programs. The motivation is type inference from PROLOG programs using SPASS. `dfg2ascii` provides an ASCII pretty print variant for SPASS input files. `dfg2dfg` is a conversion tool that transforms SPASS input files, currently dedicated to the computation/approximation of monadic clause classes (see above). A combination of the translator `dfg2tptp` with `tptp2X` yields a translation procedure from our DFG-Syntax [2] into all prover formats supported by the TPTP-library [8].

The SPASS distribution now also contains binaries for SUN Sparc Solaris, DEC Alpha Ultrix, PC Linux, PC X86 Solaris and PC Windows 95/Windows 98/Windows NT. For the Windows versions we added a neat GUI to SPASS that is built using the Qt library and also available as a SUN Sparc Solaris binary. Under a unix environment, the source code without the GUI (currently about 90000 lines) should compile without any problems if recent versions of the standard GNU tools `bison`, `flex`, `make` and `gcc` are available. The distribution is available from the SPASS homepage:

`http://spass.mpi-sb.mpg.de/`

where also links to the www-interface, documentation and problem libraries exist. The distribution of SPASS contains texinfo based documentation in different formats: man pages, info pages, html pages and postscript documents.

References

[1] T. Frühwirth, E. Shapiro, M. Y. Vardi, and E. Yardeni. Logic programs as types for logic programs. In A. R. Meyer, editor, *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science, LICS'91*, pages 300–309. IEEE Computer Society Press, July 1991.

[2] R. Hähnle, M. Kerber, and C. Weidenbach. Common syntax of the dfg-schwerpunktprogramm "deduktion". Interner Bericht 10/96, Universität Karlsruhe, Fakultät für Informatik, Germany, 1996. Current version available from `http://spass.mpi-sb.mpg.de/`.

[3] W. McCune and L. Wos. Otter. *Journal of Automated Reasoning*, 18(2):211–220, 1997.

- [4] H. Ganzinger, C. Meyer, and C. Weidenbach. Soft typing for ordered resolution. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (CADE-14)*, volume 1249 of *Lecture Notes in Computer Science*, pages 321–335, Townsville, Australia, 1997. Springer.

- [5] A. Nonnengart, G. Rock, and C. Weidenbach. On generating small clause normal forms. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 397–411, Lindau, Germany, July 1998. Springer.

- [6] C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 314–328, Trento, Italy, 1999. Springer. To appear.

- [7] C. Weidenbach, B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobalt, and D. Topic. System description: SPASS version 1.0.0. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 378–382, Trento, Italy, 1999. Springer. To appear.

[8] G. Sutcliffe and C. B. Suttner. The tptp problem library – cnf release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

10.2 Evolutionary Algorithms and its Applications in *FREAK* and *Emotion*

Investigator: Jörn Hopf

In the last two years we have been working on the applications of evolutionary algorithms (a term which collects together evolutionary programming, evolutionary strategies and genetic algorithms) to areas of optimization as diverse as non-linear predication and control, and photo mask lithography.

Evolution of Fuzzy Rules for predication and control with *FREAK*

We have been looking at how evolving algorithms can be applied to the problem of controlling, e.g., an economic system. Such systems are well known to be non-linear. In contrast to more common approaches using artificial neural networks, where the knowledge is hidden in the architecture of the network and its weighted connections, a fuzzy system provides a readable knowledge base.

We have developed the Library *FREAK* (*K*oevolutionärer *A*lgorithmus zur *E*ntwicklung von *F*uzzy-*R*eglern) for the evolution of fuzzy rules where evolutionary and coevolutionary approaches can be tested and the advantage of coevolution, which so far has been used only for function optimization, can be exploited in the development of fuzzy rules.

Our experience with this approach has been positive, and an application on steering was investigated. The current research concerns stock forecasting where a first approach shows applicable results.

Optimizing Photo Mask Layout for Grey-tone Lithography with *Emotion*

With this work we have been investigating the optimization of photo mask layout for silicon micro machining (the technology of building micro mechanical systems on silicon using photo lithography and dry-etching), in particular the problem of laying out the mask.

A drawback of the current state of this technology is that photo masks are structured as fixed arrays of 'rasters' since the general optimization problem is computationally intractable. Today, as projects grow more ambitious, this is becoming a greater problem. We have been looking at how we might be able to use an evolutionary algorithm to optimize a 'free' arrangement, unrestricted by the raster mask usually used and and the development of *Emotion* (*e*volutionary *m*ask *o*ptimiza*tion*) is in progress.

Several constraints , e.g. minimum hole size, minimum hole distance, structures of holes and placing holes side by side have to be considered. Our first results have shown that the average deviation for even a complex structure like a Fresnel-lens can be kept mostly below the wavelength of ultraviolet light. The area which can be optimized at once could be extended to $1mm^2$ and the computing time is further more reduced. Larger areas (e.g. $1cm^2$) have to be divided into parts. Because of physical effects like diffraction, we cannot optimize these parts sequentially. Hence a possible parallel/sequential approach is used.

Current work should minimize the roughness of the surface by taking advantage of the physical process of mask production and by application of a variability changing process for mutations of the evolutionary process.

(This work is part of the *Laser 2000* project of the BMBF-Program.)

10.3 COUPE: Constraint Programming and Cutting Plane Environment

Investigator: Thomas Kasper

In [1, 2], branch-and-infer, a unifying framework for integer linear programming and finite domain constraint was introduced. The COUPE system is a prototype implementation of a branch-and-infer solver. In its basic form, COUPE can be seen as a polyhedral branch-and-cut based constraint solver for pseudo-Boolean constraint programming. In comparison to other branch-and-cut solvers like CPLEX, COUPE has also the ability to solve symbolic constraints, because COUPE is an instance of the branch-and-infer framework. Among disjunctive cutting plane generation from arbitrary disjunctions, including as a special case the well-known lift-and-project method, the current version of COUPE supports symbolic constraints for expressing non-linear 0-1 inequalities, simple assignment problems, and for indicating the truth value of a linear inequality. Furthermore, COUPE has an interface to the algorithms of the OPBDP system developed by Peter Barth. COUPE is implemented in C++ and consists of about 25000 lines of code. To solve linear programs, it uses the commercial linear programming package CPLEX.

References

- [1] A. Bockmayr and T. Kasper. Branch-and-infer: A unifying framework for integer and finite domain constraint programming. *INFORMS Journal on Computing*, 10(3):287–300, 1998.
- [2] T. Kasper. *A Unifying Logical Framework for Integer Linear Programming and Finite Domain Constraint Programming.* PhD thesis, Universität des Saarlandes, December 1998.

10.4 Model Checking in CLP: a prototype

Investigators: Giorgio Delzanno, Andreas Podelski

Based on our work relating temporal properties and semantics of constraint logic programs [1], we have implemented a model checking procedure in SICStus Prolog 3.7.1 using the CLP(Q,R) library and the Boolean constraint solver (implemented with BDDs). We make extensive use of the run-time database facilities for storing and retrieving intermediate results of the analysis, and of the meta-programming facilities (e.g., the interchangeability between uninterpreted and interpreted constraints expressions).

We have applied the implementation to several infinite-state verification problems that are becoming benchmarks in the community (e.g. mutual-exclusion algorithms). This allowed us to evaluate the performance of our implementation, to experiment with evaluation strategies and abstractions, and to compare our solution with previous solutions.

We implement the solving of constraints over integers, which is needed for model checking integer-valued concurrent systems, through a constraint solver over reals. We thus trade the theoretical and practical gain in efficiency with an extra abstraction.

References

- [1] G. Delzanno and A. Podelski. Model Checking in CLP. In R. Cleaveland, editor, *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems*

(TACAS'99) held as part of ETAPS'99, volume 1579 of *Lecture Notes in Computer Science*, pages 223–239, Amsterdam, The Netherlands, January 1999. European Association for Programming Languages and Systems (EAPLS); European Association for Theoretical Computer Science (EATCS), Springer.

11 Journal and conference activities

11.1 Editorial positions

The editorial activities of staff of the programming logics group are as follows:

Harald Ganzinger is an editor of the following journals:

- *Information Processing Letters*
- *Journal of Automated Reasoning*
- *Theory of Computing Systems*
- *Discrete Mathematics and Computer Science*

Andreas Nonnengart was co-editor of the proceedings of the first *International Joint Conference on Qualitative and Quantitative Practical Reasoning* (ECSQARU/FAPR'97).

Andreas Podelski is an editor of the *International Journal on Software Tools for Technology Tranfer*

11.2 Conference activities

The participation of the members of the programming logics group in the organisation of various conferences is as follows.

Program comittees

Giorgio Delzanno was a member of the program comittee of the *Post-Conference Workshop on Transactions and Change in Logic Databases* (DYNAMICS'98) at the Joint International Conference and Symposium on Logic Programming.

Harald Ganzinger is or was a member of the program comittee of

- the 1999 *International Conference on Automated Deduction* (CADE-16) (as chair)
- the *6th International Conference on Logic for Programming and Automated Reasoning* (LPAR'99) (as co-chair)
- the 1999 conference on *Foundations of Software Science and Computation Structures* (FOSSACS'99)
- the 1999 *Annual Conference of the European Association for Computer Science Logic* (CSL'99)
- the 1998 *International Conference on Automated Deduction* (CADE-15)
- the 1998 conference on *Rewriting Techniques and Applications* (RTA'98)
- the 1997 *International Conference on Automated Deduction* (CADE-14)
- the 1997 conference on *Rewriting Techniques and Applications* (RTA'97)
- the 1997 conference on *Principles of Programming Languages* (POPL'97)

Andreas Nonnengart was a member of the program comittee of the *First International Joint Conference on Qualitative and Quantitative Practical Reasoning* (ECSQARU/FAPR'97).

Andreas Podelski is or was a member of the program comittees of

- the 1999 *International Symposium on Static Analysis* (SAS'99)

- the *Fourth International Workshop on Set Constraints and Constraint-based Program Analysis* (Set Constraints'98)
- the *First International Conference on Foundations of Software Science and Computation Structures* (FOSSACS'98, part of the *European Joint Conferences on Theory and Practice of Software* (ETAPS))
- the *Third International Conference on Principles and Practice of Constraint Programming* (CP97)
- the *First International Workshop on Concurrent Constraint Programming for Time Critical Applications* (COTIC 97)
- the *Third International Workshop on Set Constraints and Constraint-based Program Analysis* (Set Constraints'97)
- the *Fourteenth International Logic Programming Symposium* (ILPS'97),

Emil Weydert was a member of the program committee of

- the *Fourth Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications* (DGNMR 99)
- the *First International Joint Conference on Qualitative and Quantitative Reasoning* (ECSQARU/FAPR 97)
- the *Third Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications* (DGNMR 97)

Organisation of workshops and conferences

Andreas Nonnengart was a member of the organising comittee of the *First International Joint Conference on Qualitative and Quantitative Practical Reasoning* (ECSQARU/FAPR'97).

Andreas Podelski was on the organising commitee of the *Dagstuhl Seminar on Concurrent Constraint Programming.* October 6 - 10, 1997.

Christoph Weidenbach is an elected member of the steering committee of the *International Workshop on First Order Theorem Proving* (FTP).

Emil Weydert was the organiser of the *Third Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications* (DGNMR 97)

12 Teaching Activities

Apart from the core Computer Science course on programming languages, the group also contributes a wide range of general and specialist lectures and seminars to the logic and computation curriculum organised together with DFKI and the Computer Science Department. (Unless specified, courses were taught at the University of Saarbrücken.)
Key: L – Lectures, LE – Lectures and exercises, E – Exercises, S – Seminar,
FoPra – Project class.

Summer Semester 1997

Constraintprogramming A. Bockmayr – L (at the University of Wroclaw)
Informatik IV H. Ganzinger – LE
Praxis des Contraintprogrammierens A. Bockmayr – LE
Verifikation H. Ganzinger, A. Podelski – S
Set constraints, their use for program analysis and for solving constraint problems over (feature) trees A. Podelski – L (ESSLLI'97 in Aix-en-Provence)

Winter Semester 1997/1998

Anwendungen der Constraintprogrammierung A. Bockmayr – FoPra
Verifikation verteilter Systeme H. Ganzinger, A. Podelski – S
Theoretische Grundlagen der Objektorientierung S. Matthews – L
Praxis des Programmierens C. Weidenbach – LE
Probabilistisches Schließen E. Weydert – L

Summer Semester 1998

Optimierung A. Bockmayr – LE
Unix für fortgeschrittene Benutzer U. Waldmann – LE
Rechnergestütztes Beweisen A. Podelski, C. Weidenbach – LE
Schließen und Entscheiden mit graphischen und logischen Modellen H. Ganzinger, M. Jaeger, E. Weydert – S

Winter Semester 1998/1999

Logik für Informatiker H. Ganzinger – LE
Formale Methoden in der Sicherheit H. Ganzinger, B. Pfitzmann, A. Podelski, M. Schunter – S
Optimierung A. Bockmayr, F. Eisenbrand – S
Formal systems of probabilistic inference M. Jaeger, E. Weydert – L (ESSLLI'99 in Saarbrücken)
Automated Reasoning C. Weidenbach – LE (at the International Masters Program at the University of Dresden)

Summer Semester 1999

Termersetzungssysteme H. Ganzinger – LE
Theorie und Anwendung Bayesscher Netzwerke und verwandter Formalismen M. Jaeger, A. Jameson – LE
Einführung für die Hörer aller Fakultäten A. Podelski – LE
Universelle Algebra und Verbandstheorie V. Sofronie-Stokkermans – LE
Praxis des Programmierens U. Waldmann – LE
Beweisen mit SPASS C. Weidenbach – FoPra

Diploma students

Members of the programming logics group were the advisors for the following diploma theses of students of the University of the Saarland.

Michael Christen, *A Calculus of Simplification for Superposition*
Stefan Friedrich, *Integration of a Decision Procedure for Second-Order Monadic Logic in a Higher-Order Logic Theorem Proving Environment*
Peter Leven, *Integrating Clausal Decision Procedures in a Tactic Based Theorem Prover*
Jan Timm, *Testing the Satisfiability of RPO Constraints*

13 Dissertations and Habilitations

13.1 Doctorates

Completed

U. Waldmann, *Cancellative Abelian Monoids in Refutational Theorem Proving*, (July 1997)
L. Viganò, *A Framework for Non-Classical Logics*, (September 1997)
R.A. Schmidt, *Optimised Modal Translation and Resolution*, (November 1997)
T. Kasper, *A Unifying Logical Framework for Integer Linear Programming and Finite Domain Constraint Programming*, (December 1998)
G. Struth, *Canonical Transformations in Algebra, Universal Algebra and Logic*, (April 1999)

In Progress

H. Baumeister, *Relations between Abstract Datatypes modeled as Abstract Datatypes* (submitted)
U. Hustadt, *Resolution-Based Decision Procedures for Subclasses of First-Order Logic* (submitted)
J. Hopf, *Combinatorial Optimization of Photo Mask Layout for Grey-tone Lithography by an Evolutionary Algorithm*
J. Stuber *Superposition Theorem Proving for Algebraic Theories*
M. Tzakova, *Hybrid Languages*
P. Maier, *Constraint-based Compositional Verification*
C. Meyer, *Soft Typing for Clausal Inference Systems*

13.2 Habilitations

Completed

A. Podelski

In Progress

S. Matthews

14 Grants and cooperations

Theorembeweisen und Algebra

Description

The commonest techniques of automated theorem proving (e.g. resolution) are not well suited for working with typical algebraic structures, since the large number of inferences possible at any time (because of, e.g., associativity, commutativity or transitivity) means that that the search space can grow uncontrollably.

One possible solution to this problem is to integrate the algebraic axioms into the theorem prover itself, where we can better control their application according to the particular circumstances. The search space can be further controlled by exploiting quantifier elimination techniques and the redundancy criteria of the theory.

The purpose of this project was to exploit and further develop these techniques for new algebraic structures (e.g. Abelian monoids or groups with further properties such as torsion-freeness or partial or total ordering).

Technical Data

Starting date:	September 1, 1996
Duration:	2 years (completed)
Funding:	DFG
Staff at MPI f. Informatik:	Harald Ganzinger Jürgen Stuber Uwe Waldmann

Partners

The project was part of the "Schwerpunktprogramm Deduktion", the partners of which included: Universität Braunschweig; Universität Ulm; Universität Kaiserslautern; Universität Berlin.

Steuerung der Beweissuche durch Abstraktion

Description

The aim of this project was to investigate abstraction techniques that allow us to direct the search for a proof. We have proposed a variant of ordered resolution with semantic restrictions based on interpretations which are identified by the given atom ordering and selection function. Techniques for automatically and effectively approximating validity (satisfiability) via abstraction in these interpretations are presented and related to methods of soft typing for programming languages. The abstracted interpretations are then used to detect redundant clauses and to select inferences. The framework is shown to be strictly more general than certain previously introduced approaches. Implementation of some of our techniques in the SPASS(see §10.1) prover has lead to encouraging experimental results.

Technical Data

Starting date:	September 1, 1996
Duration:	2 years (completed)
Funding:	DFG
Staff at MPI f. Informatik:	Harald Ganzinger Ullrich Hustadt Christoph Meyer Christoph Weidenbach

Partners

The project was part of the 'Schwerpunktprogramm Deduktion', the partners of which included: Universität Braunschweig; Universität Ulm; Universität Kaiserslautern; Universität Berlin.

CONSOLE: Constraint Solving in Europe

Description

The goal of this project was to facilitate the interactions between European research teams in the field of constraint solving, especially concerning visits and exchanges of young researchers. The work mainly focused on symbolic constraints (i.e. logic formulae interpreted in some tree structure) and on the application of constraints to constraint logic programming languages.

The MPI-part in this project is mainly concerned with

- Paramodulation and Superposition Calculi
- Set Constraints and the Monadic Class
- Non-Linear Constraints in CLP($\mathcal{R}$)
- 0-1 Constraints in CLP($\mathcal{PB}$)

Technical Data

Starting date:	1 January 1995
Duration:	3 years (completed)
Funding:	Human Capital and Mobility
Staff at MPI f. Informatik:	Peter Barth Alexander Bockmayr Witold Charatonik Harald Ganzinger Andreas Podelski Uwe Waldmann

Partners

University of Barcelona; University of Lille; INRIA Lorraine; University of Orsay; Cosytec, Orsay; University of Padova.

CCL II: Construction of Computational Logics II

Description

The objectives of CCL II are

- to investigate specific instances of combination problems for logics and constraints of particular interest
- to investigate new symbolic constraints and to design algorithms for combining existing constraint systems
- to develop or improve theorem proving techniques for certain logics of special importance for programming, by taking advance of constraint systems.

Technical Data

Starting date:	September 1996
Duration:	3 years
Funding:	ESPRIT Basic Research Working Group 22457
Staff at MPI f. Informatik:	Harald Ganzinger Alexander Bockmayr Witold Charatonik Thomas Kasper Andreas Podelski Sergei Vorobyov Uwe Waldmann

Partners

CIS, Univ. München, Germany. COSYTEC, Orsay, France. DFKI Saarbrücken, Germany. Hebrew Univ., Jerusalem, Israel. INRIA Nancy, France. LIFL, Lille, France. LRI, Univ. Paris-Sud, France. RWTH Aachen, Germany. TU München, Germany. UCM, Madrid, Spain. Univ. Frankfurt, Germany. Univ. Wroclaw, Poland. UPC, Barcelona, Spain

Defaults and Probability (DEPRO)

Description

This project is concerned with the investigation of probabilistic and quasi-probabilistic approaches to default reasoning. These accounts offer a transparent semantics and support plausible inferences in accordance with our commonsense intuitions. Major issues are the handling of first-order default knowledge expressed by genuine default quantifiers and the relationship with probabilistic reasoning strategies like entropy maximization.

Technical Data

Starting date:	May 1, 1996
Duration:	3 years
Funding:	DFG
Staff at MPI f. Informatik:	Richard Booth Harald Ganzinger Manfred Jaeger Leon van der Torre Emil Weydert

Partners

University of Konstanz, University of Toulouse, Stanford University

15 Publications

Books and monographs

[1] D. Gabbay, R. Kruse, A. Nonnengart, and H. J. Ohlbach, editors. *Qualitative and Quantitative Practical Reasoning*, volume 1244 of *Lecture Notes in Artificial Intelligence, LNAI*. Springer, Berlin, Germany, 1997. Proceedings of the First International Joint Conference on Qualitative and Quantitative Practical Reasoning, ECSQARU/FAPR'97, Bad Honnef, Germany.

[2] E. Weydert, G. Brewka, and C. Witteveen, editors. *Proceedings of the Third Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR-97)*, Saarbrücken, Germany, 1997. Max-Planck-Institut für Informatik.

Journals and chapters

[1] L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction: A Basis for Applications*, volume I, chapter 11, pages 353–397. Kluwer, Dordrecht, Netherlands, 1998.

[2] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998. Revised Version of MPI-I-95-2-009.

[3] L. Bachmair and H. Ganzinger. A theory of resolution. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, 1999. To appear.

[4] P. Barth and A. Bockmayr. Modelling discrete optimisation problems in constraint logic programming. *Annals of Operations Research*, 81:467–496, 1998.

[5] D. A. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

[6] D. A. Basin, S. Matthews, and L. Viganò. Labelled modal logics: quantifiers. *Journal of Logic, Language and Information*, 7:237–263, 1998.

[7] D. A. Basin, S. Matthews, and L. Viganò. Natural deduction for non-classical logics. *Studia Logica*, 60(1):119–160, 1998.

[8] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6(4):625–650, 1998. Revised Version of MPI-I-97-2-007.

[9] P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):23–49, 1998.

[10] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999. Revised Version of MPI-I-98-2-006.

[11] A. Bockmayr, F. Eisenbrand, M. Hartmann, and A. S. Schulz. On the Chvátal rank of polytopes in the 0/1 cube. *Discrete Applied Mathematics*. To appear.

[12] A. Bockmayr and T. Kasper. Branch-and-infer: A unifying framework for integer and finite domain constraint programming. *INFORMS Journal on Computing*, 10(3):287–300, 1998.

[13] M. Bozzano, G. Delzanno, M. Martelli, V. Mascardi, and F. Zini. Logic Programming and Multi-Agent Systems: a Synergic Combination for Applications and Semantics. In *The Logic Programming Paradigm: a 25-Year Perspectiv*, Lecture Notes in Computer Science, chapter 1, pages 5–32. Springer, Berlin, 1999. To appear.

[14] J. Buchmann and F. Eisenbrand. On factor refinement in number fields. *Mathematics of Computation*, 68(225):345–350, 1999.

[15] M. Bugliesi, G. Delzanno, L. Liquori, and M. Martelli. Object Calculi in Linear Logic. *Journal of Logic and Computation*, 1999. To appear.

[16] W. Charatonik. Set constraints in some equational theories. *Information and Computation*, 142:40–75, 1998.

[17] W. Charatonik. An undecidable fragment of the theory of set constraints. *Information Processing Letters*, 68:147–151, 1998.

[18] W. Charatonik and A. Podelski. Set constraints with intersection. *Information and Computation*, 1999. To appear.

[19] A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. Decidability and complexity of simultaneous rigid e-unification with one variable and related results. *Theoretical Computer Science*, 1999. To appear.

[20] G. Delzanno, D. Galmiche, and M. Martelli. A Specification Logic for Concurrent Object-oriented Programming. *Mathematical Structures in Computer Science*, 1999. To appear.

[21] F. Eisenbrand. Note on the membership problem for the elementary closure of a polyhedron. *Combinatorica*. To appear.

[22] H. Ganzinger, U. Hustadt, C. Meyer, and R. A. Schmidt. A resolution-based decision procedure for extensions of K4. In *Advances in Modal Logic, Volume 2*, CSLI Lecture Notes. CSLI, Stanford, USA, June 1999. To appear.

[23] Y. Gurevich and M. Veanes. Logic with equality: Partisan corroboration, and shifted pairing. *Information and Computation*, 1999. To appear.

[24] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, page 42, 1999. To appear.

[25] M. R. K. Krishna Rao. A framework for incremental learning of logic programs. *Theoretical Computer Science*, 185:191–213, 1997.

[26] M. R. K. Krishna Rao. Modular aspects of term graph rewriting. *Theoretical Computer Science*, 208(1-2):59–86, 1998.

[27] R. Letz and C. Weidenbach. Paradigmen und Perspektiven der automatischen Deduktion. *KI, Organ des Fachbereichs 1 "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.*, 4:15–19, 1998.

[28] J. Levy and M. Veanes. On the undecidability of second-order unification. *Information and Computation*, 1999. To appear.

[29] M. Lu and J. Wu. On theorem proving in annotated logics. *Journal of Applied Non-Classical Logics*, 1999. To appear.

[30] S. Matthews and D. A. Basin. Scoped metatheorems. *Electronic Notes in Computer Science*, 15:1–14, 1998.

[31] S. Matthews and D. A. Basin. Structuring metatheory on inductive definitions. *Information and Computation*, 1999. To appear.

[32] M. Müller, J. Niehren, and A. Podelski. Ordering constraints over feature trees. *Constraints*, 1999. To appear.

[33] M. Nivat and A. Podelski. Minimal ascending and descending tree automata. *SIAM Journal on Computing*, 26(1):39–58, 1997.

[34] A. Nonnengart. Modal frame characterization by way of auxiliary modalities. *Logic Journal of the IGPL*, 6(6):875–899, 1998.

[35] A. Nonnengart, H. J. Ohlbach, and D. Gabbay. Encoding two-valued non-classical logics in classic logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, Netherlands, 1999. To appear.

[36] A. Nonnengart, H. J. Ohlbach, and A. Szalas. Quantifier elimination for second-order predicate logic. To appear in *Logic, Language and Reasoning. Essays in honour of Dov Gabbay, Part I, Kluwer Academic Press*, 1999.

[37] A. Nonnengart and A. Szalas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In E. Orlowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, volume 24 of *Studies in Fuzziness and Soft Computing*, page 18. Springer, Berlin, 1999.

[38] A. Nonnengart and C. Weidenbach. On generating small clause normal forms. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, Netherlands, 1999. To appear.

[39] H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *Journal of Logic and Computation*, 7(5):581–603, 1997. Also available as Research Report MPI-I-95-2-002.

[40] A. Podelski and G. Smolka. Situated simplification. *Theoretical Computer Science*, 173(1):235–252, 1997. Preliminary Version in Ugo Montanari, editor, Proceedings of the First International Conference on Principles and Practice of Constraint Programming (CP'95). Springer LNCS 976, 1995.

[41] R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, volume 87 of *CSLI Lecture Notes*, chapter 13, pages 189–208. CSLI, Stanford, USA, 1998.

[42] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.

[43] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators and applications to the semantics of non-classical logics. part i. *Studia Logica*, 1999. To appear.

[44] V. Sofronie-Stokkermans. Duality and canonical extensions of bounded distributive lattices with operators and applications to the semantics of non-classical logics. part ii. *Studia Logica*, 1999. To appear.

[45] V. Sofronie-Stokkermans. Priestley duality for SHn-algebras and applications to the study of Kripke-style models for SHn-logics. *Multiple-Valued Logic - An International Journal*, 1999. To appear.

[46] J. Stuber. Superposition theorem proving for abelian groups represented as integer modules. *Theoretical Computer Science*, 208(1-2):149–177, 1998.

[47] J. Stuber. Superposition theorem proving for commutative rings. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction - A Basis for Applications. Volume III. Applications*, volume 10 of *Applied Logic Series*, chapter 2, pages 31–55. Kluwer, Dordrecht, Netherlands, 1998.

[48] J.-M. Talbot, P. Devienne, and S. Tison. Generalized definite set constraints. *CONSTRAINTS - An International Journal*, 5, 1999. To appear.

[49] L. van der Torre and Y.-H. Tan. An update semantics for deontic reasoning. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, volume 49 of *Frontiers in Artificial Intelligence and Applications*, pages 73–90. IOS, Amsterdam, 1999.

[50] L. W. N. van der Torre and Y.-H. Tan. Contextual deontic logic: violation contexts and factual defeasibility. In M. Cavalcanti, editor, *Formal Aspects in Context*, Applied Logic Series. Kluwer, Dordrecht, 1999.

[51] L. W. N. van der Torre and Y.-H. Tan. Diagnosis and decision making in normative reasoning. *Journal of artificial intelligence and law*, 1999. To appear.

[52] U. Waldmann. Extending reduction orderings to ACU-compatible reduction orderings. *Information Processing Letters*, 67(1):43–49, 1998.

[53] C. Weidenbach. SPASS version 0.49. *Journal of Automated Reasoning*, 18(2):247–252, 1997.

[54] C. Weidenbach. Rechnen in sortierter Prädikatenlogik. In H. Fiedler, P. Gorny, W. Grass, S. Hölldobler, G. Hotz, I. O. Kerner, and R. Reischuk, editors, *Ausgezeichnete Informatikdissertationen 1997*, pages 183–197. Teubner, Stuttgart, Germany, 1998.

[55] C. Weidenbach. Sorted unification and tree automata. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction - A Basis for Applications*, Applied Logic, chapter 9, pages 291–320. Kluwer, Dordrecht, Netherlands, 1998.

[56] C. Weidenbach. Spass: Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, Amsterdam, Netherlands, 1999. To appear.

[57] C. Weidenbach, C. Meyer, C. Cohrs, T. Engel, and E. Keen. Spass v0.77. *Journal of Automated Reasoning*, 21(1):113–113, 1998.

[58] A. Werner, A. Bockmayr, and S. Krischer. How to realize LSE narrowing. *New Generation Computing*, 16(4):397–434, 1998.

[59] J. Wu. First-order polynomial based theorem proving. In X.-S. Gao and D. Wang, editors, *Mathematics Mechanizations and Applications*, page 21. Academic, London, 1999.

[60] J. Wu and Z. Liu. Well-behaved inference rules for first-order theorem proving. *Journal of Automated Reasoning*, 21(3):381–400, 1998.

[61] J. Wu, H. Tan, and Y. Li. An algebraic method to decide the deduction problem in many-valued logics. *Journal of Applied Non-Classical Logics*, 8(4):353–360, 1998.

[62] V. Sofronie-Stokkermans. Automated theorem proving by resolution for finitely-valued logics based on distributive lattices with operators. *Multiple-Valued Logic - An International Journal*, 1999. To appear.

Conference proceedings

[1] L. Bachmair and H. Ganzinger. Strict basic superposition. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 160–174, Lindau, Germany, July 1998. Springer.

[2] L. Bachmair, H. Ganzinger, and A. Voronkov. Elimination of equality via transformation with ordering constraints. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 175–190, Lindau, Germany, July 1998. Springer. Short version of MPI-I-97-2-012.

[3] P. Barth and A. Bockmayr. PLAM: Prolog and algebraic modelling. In *Proceedings of the Fifth International Conference on the Practical Application of Prolog*, pages 73–82, London, UK, 1997. The Practical Application Company.

[4] D. A. Basin, S. Matthews, and L. Viganò. Labelled quantified modal logics. In G. Brewka, C. Habel, and B. Nebel, editors, *Proceedings of the 21st Annual German Conference on Artificial Intelligence (KI-97): Advances in Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 171–182, Freiburg, Germany, 1997. Springer.

[5] D. A. Basin, S. Matthews, and L. Viganò. A new method for bounding the complexity of modal logics. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory (KGC-97)*, volume 1289 of *Lecture Notes in Computer Science*, pages 89–102, Vienna, Austria, 1997. Springer.

[6] D. A. Basin, S. Matthews, and L. Viganò. A modular presentation of modal logics in a logical framework. In J. Ginzburg, Z. Khasidashvili, C. Vogel, J.-J. Levy, and E. Vallduvi, editors, *Proceedings of the 1st Tbilisi Symposium on Language, Logic and Computation: Selected Papers*, Studies in Logic, Langugage and Information, pages 293–307, Tbilisi, Georgia, 1998. folli, CSLI.

[7] P. Blackburn and M. Tzakova. A hybrid concept language. In E. Boros and R. Greiner, editors, *Proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida, 1998. `http://rutcor.rutgers.edu/~amai`.

[8] M. Bozzano, G. Delzanno, M. Martelli, V. Mascardi, and F. Zini. Multi-Agent Systems Development as a Software Engineering Enterprise. In G. Gupta, editor, *Proceedings of the First International Workshop on Practical Aspects of Declarative Languages (PADL 99)*, volume 1551 of *Lecture Notes in Computer Science*, pages 46–60, San Antonio, Texas, USA, October 1998. Compulog Americas and Association for Logic Programming (ALP), Springer.

[9] W. Charatonik, D. McAllester, D. Niwinski, A. Podelski, and I. Walukiewicz. The horn mu-calculus. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 58–69, Indianapolis, Indiana, June 1998. IEEE Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.

[10] W. Charatonik and A. Podelski. Set constraints with intersection. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97)*, pages 362–372, Warsaw, Poland, 1997. IEEE Comput. Soc. Press.

[11] W. Charatonik and A. Podelski. Co-definite set constraints. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 211–225, Tsukuba, Japan, March 1998. Springer.

[12] W. Charatonik and A. Podelski. Directional type inference for logic programs. In G. Levi, editor, *Proceedings of the 5th International Symposium in Static Analysis (SAS-98)*, volume 1503 of *Lecture Notes in Computer Science*, pages 278–294, Pisa, Italy, September 1998. Springer.

[13] W. Charatonik and A. Podelski. Set-based analysis of reactive infinite-state systems. In B. Steffen, editor, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS-98)*, volume 1384 of *Lecture Notes in Computer Science*, pages 358–375, Lisbon, Portugal, March 1998. Springer.

[14] V. Cortier, H. Ganzinger, F. Jacquemard, and M. Veanes. Decidable fragments of simultaneous rigid reachability. In *Proc. ICALP'99*, 1999. To appear. Full version of this paper is available as MPI Research Report MPI-I-1999-2-004.

[15] A. Degtyarev, Y. Gurevich, P. Narendran, M. Veanes, and A. Voronkov. The decidability of simultaneous rigid e-unification with one variable. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 181–195, Tsukuba, Japan, March 1998. Springer.

[16] G. Delzanno and A. Podelski. Model Checking in CLP. In R. Cleaveland, editor, *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'99) held as part of ETAPS'99*, volume 1579 of *Lecture Notes in Computer Science*, pages 223–239, Amsterdam, The Netherlands, January 1999. European Association for Programming Languages and Systems (EAPLS); European Association for Theoretical Computer Science (EATCS), Springer.

[17] G. Delzanno and A. Podelski. Verification of infinite-state systems in constraint logic programming. In F. Fages, editor, *Proceedings of the French Conference of Logic Programming*, March 1999. To appear.

[18] F. Eisenbrand and A. S. Schulz. Bounds on the Chvátal rank of polytopes in the 0/1-cube. In *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, Graz, Austria, 1999. Springer. To appear.

[19] B. S. Firozabadhi and L. W. N. van der Torre. Towards a formal analysis of control systems. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 317–318, Brighton, England, 1998. Wiley.

[20] H. Ganzinger and H. De Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999. To appear.

[21] H. Ganzinger, F. Jacquemard, and M. Veanes. Rigid reachability. In J. Hsiang and A. Ohori, editors, *Proceedings of the 4th Asian Computing Science Conference on Advances in Computing Science (ASIAN-98)*, volume 1538 of *Lecture Notes in Computer Science*, pages 4–21, Manila, The Philippines,, 1998. Springer. A full version of this paper is available as MPI-I Research Report MPI-I-98-2-013.

[22] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999. To appear.

[23] H. Ganzinger, C. Meyer, and C. Weidenbach. Soft typing for ordered resolution. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (CADE-14)*, volume 1249 of *Lecture Notes in Computer Science*, pages 321–335, Townsville, Australia, 1997. Springer.

[24] J. Hopf. Cooperative coevolution of fuzzy rules. In N. Steele, editor, *Proceedings of the 2nd International ICSC Symposium on Fuzzy Logic and Applications (ISFL-97)*, pages 337–381, Zurich, Switzerland, February 1997. International Computer Science Conventions (ICSC), ICSC Academic Press.

[25] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logics. In M. E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 202–207, Nagoya, Japan, 1997. International Joint Conferences on Artificial Intelligence, Inc. (IJCAII) and Japanese Society for Artificial Intelligence (JSAI), Morgan Kaufmann.

[26] U. Hustadt and R. A. Schmidt. Simplification and backjumping in modal tableau. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reaso ning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 187–201, Oisterwijk, The Netherlands, May 1998. Springer.

[27] U. Hustadt and R. A. Schmidt. Maslov's class k revisited. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, Lecture Notes in Artificial Intelligence Series:, Trento, Italy, 1999. Springer. To appear.

[28] U. Hustadt and R. A. Schmidt. On the relation of resolution and tableaux proof systems for description logics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, 1999. Morgan Kaufmann. To appear.

[29] U. Hustadt, R. A. Schmidt, and C. Weidenbach. Optimised functional translation and resolution. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reaso ning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 36–37, Oisterwijk, The Netherlands, May 1998. Springer.

[30] F. Jacquemard, C. Meyer, and C. Weidenbach. Unification in extensions of shallow equational theories. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 76–90, Tsukuba, Japan, 1998. Springer.

[31] M. Jaeger. Relational bayesian networks. In D. Geiger and P. P. Shenoy, editors, *Proceedings of the 13th Conference of Uncertainty in Artificial Intelligence (UAI-13)*, pages 266–273, Providence, USA, 1997. Morgan Kaufmann.

[32] M. Jaeger. Convergence results for relational Bayesian networks. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 44–55, Indianapolis, USA, 1998. IEEE Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.

[33] M. Jaeger. Measure selection: Notions of rationality and representation independence. In G. S. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 274–281, Madison, USA, 1998. Morgan Kaufmann.

[34] M. Jaeger. Reasoning about infinite random structures with relational bayesian networks. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pages 570–581, Trento, Italy, 1998. Morgan Kaufmann.

[35] S. Matthews. Extending a logical framework with a modal connective for validity. In M. Abadi and T. Ito, editors, *Proceedings of the 3rd Symposium on Theoretical Aspects of Computer Software (TACS-97)*, volume 1281 of *Lecture Notes in Computer Science*, pages 491–514, Sendai, Japan, September 1997. Tohoku University, ACM SIGACT, the Information Processing Society of Japan, the Japan Society of Software Science, Association for Symbolic Logic, Springer.

[36] S. Matthews. A practical implementation of simple consequence relations using inductive definitions. In W. McCune, editor, *Proceedings of the 14th International Conference on Automated Deduction (CADE-14)*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 306–320, Townsville, Australia, July 1997. Association for Automated Reasoning, Springer.

[37] M. Müller, J. Niehren, and A. Podelski. Inclusion constraints over non-empty sets of trees. In M. Bidoit and M. Dauchet, editors, *Proceedings of the 7th International Joint Conference CAAP/FASE: Theory and practice of software development (TAPSOFT-97)*, volume 1214 of *Lecture Notes in Computer Science*, pages 345–356, Lille, France, April 1997. Springer.

[38] M. Müller, J. Niehren, and A. Podelski. Ordering constraints over feature trees. In G. Smolka, editor, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP-97)*, volume 1330 of *Lecture Notes in Computer Science*, pages 549–562, Linz, Austria, 1997. Springer.

[39] A. Nonnengart, G. Rock, and C. Weidenbach. On generating small clause normal forms. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 397–411, Lindau, Germany, July 1998. Springer.

[40] L. Pacholski and A. Podelski. Set constraints: a pearl in research on constraints. In G. Smolka, editor, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP-97)*, volume 1330 of *Lecture Notes in Computer Science*, pages 549–562, Linz, Austria, 1997. Springer.

[41] A. Podelski. Set-based analysis of logic programs and reactive logic programs. In J. Maluszynski, editor, *Proceedings of the International Symposium on Logic Programming (ILPS-14)*, pages 35–36, Leuven, Belgium, 1997. MIT Press.

[42] A. Podelski, W. Charatonik, and M. Müller. Set-based failure analysis for logic programs and concurrent constraint programs. In S. D. Swierstra, editor, *Programming Languages and Systems. Proceedings of the 8th European Symposium on Programming ESOP'99*, volume 1576 of *Lecture Notes in Computer Science*, pages 177–192, Amsterdam, The Netherlands, March 1999. Springer.

[43] R. A. Schmidt. E-unification for subsystems of S4. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA-98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 106–120, Tsukuba, Japan, 1998. Springer.

[44] V. Sofronie-Stokkermans. On translation of finitely-valued logics to classical first-order logic. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 410–411, Brighton, UK, 1998. Wiley.

[45] V. Sofronie-Stokkermans. On the universal theory of varieties of distributive lattices with operators: Some decidability and complexity results. In *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, Lecture Notes in Computer Science, Trento, Italy, 1999. Springer. To appear.

[46] V. Sofronie-Stokkermans. Representation theorems and automated theorem proving in non-classical logics. In *Proceedings of the 29th IEEE International Symposium on Multiple-Valued Logic (ISMVL 99)*, Freiburg im Breisgau, Germany, 1999. IEEE Computer Society Press. To appear.

[47] G. Struth. On the word problem for free lattices. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications (RTA-97)*, volume 1103 of *Lecture Notes in Computer Science*, pages 128–141, Sitges, Spain, June 1997. Springer.

[48] J. Stuber. Theory path orderings. In *Tenth International Conference on Rewriting Techniques and Applications (RTA99)*, Lecture Notes in Computer Science, Trento, Italy, 1999. Springer. To appear.

[49] J.-M. Talbot, J. Niehren, and M. Müller. Entailment of atomic set constraints is pspace-complete. In G. Longo, editor, *Proceedings of the 14th Annual IEEE Symposium on logic in computer science (LICS'99)*, Trento, Italy, 1999. IEEE Comput. Soc. Press. To appear.

[50] M. Tzakova. Tableau calculi for hybrid logics. In N. V. Murray, editor, *Proceedings of the International Conference TABLEAUX'99 - Analytic Tableaux and Related Methods*, LNAI, Saratoga Springs, NY USA, 1999. Springer. To appear.

[51] L. van der Torre and Y.-H. Tan. An update semantics for defeasible obligations. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, 1999. To appear.

[52] L. W. N. van der Torre. Labeled logics of conditional goals. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 368–369, Brighton, England, 1998. Wiley.

[53] L. W. N. van der Torre. Defeasible goals. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Proceedings of the ECSQARU'99*, LNAI, London, 1999. Springer. To appear.

[54] L. W. N. van der Torre and Y.-H. Tan. Distinguishing different roles in normative reasoning. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL-97)*, pages 225–232, Melbourne, Australia, 1997. The international association for artificial intelligence and law; the university of Melbourne law school, ACM Press.

[55] L. W. N. van der Torre and Y.-H. Tan. Reasoning about exceptions. In G. Brewka, C. Habel, and B. Nebel, editors, *Proceedings of the 21st Annual German Conference on Artificial Intelligence (KI-97): Advances in Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 405–408, Freiburg, September 1997. Springer.

[56] L. W. N. van der Torre and Y.-H. Tan. The temporal analysis of Chisholm's paradox. In *Proceedings of the 15th National Conference, and 10th Conference on Innovative Applications of Artificial Intelligence (AAAI-98) and (IAAI-98)*, pages 650–655, Madison, Wisconson, 1998. American Association for Artificial Intelligence (AAAI), AAAI Press/ MIT Press.

[57] L. W. N. van der Torre and Y.-H. Tan. An update semantics for prima facie obligations. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 38–42, Brighton, England, 1998. Wiley.

[58] L. W. N. van der Torre and Y.-H. Tan. Rights, duties and commitments between agents. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, 1999. IJCAII and the Scandinavian AI Societies, Morgan Kaufmann. To appear.

[59] M. Veanes. The relation between second-order unification and simultaneous rigid e-unification. In V. Pratt, editor, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS-98)*, pages 264–275, Indianapolis, Indiana, 1998. IEEE Computer Society Technical Committee on Mathematical Foundations of Computing, IEEE Computer Society Press.

[60] S. Vorobyov. The first-order theory of one step rewriting in linear noetherian systems is undecidable. In H. Comon, editor, *Proceedings of the 8th International Conference on Rewriting Techniques and Applications (RTA-97)*, volume 1232 of *Lecture Notes in Computer Science*, pages 254–268, Sitges, Spain, June 1997. Springer.

[61] S. Vorobyov. The 'hardest natural decidable theory. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS-97)*, pages 294–305, Warsaw, Poland, June-July 1997. IEEE Comput. Soc. Press.

[62] S. Vorobyov. $\forall\exists^*$-equational theory of context unification is Π_1^0-hard. In L. Brim, J. Gruska, and J. Zlatuska, editors, *Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS-98)*, volume 1450 of *Lecture Notes in Computer Science*, pages 597–606, Brno, Czech Republic, August 1998. Mazaryk University, European Assoc of Computer Science Logic, Springer.

[63] S. Vorobyov. Subtyping functional+nonempty record types. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic (CSL'98)*, volume 1584 of *Lecture Notes in Computer Science*, pages 280–295, Brno, Czech Republic, 1999. European Association for Computer Science Logic, Springer.

[64] S. Vorobyov and A. Voronkov. Complexity of nonrecursive logic programs with complex values. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-98)*, pages 244–253, Seattle, Washington, U.S.A., June 1998. Assoc. Comput. Machinery, Boeing, ACM Press.

[65] U. Waldmann. A superposition calculus for divisible torsion-free abelian groups. In M. P. Bonacina and U. Furbach, editors, *Proceedings of the International Workshop on First-Order Theorem Proving (FTP-97)*, number 97-50 in RISC-Linz Report Series, pages 130–134, Linz, Austria, 1997. Johannes Kepler Universität.

[66] U. Waldmann. Superposition for divisible torsion-free abelian groups. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-98)*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 144–159, Lindau, Germany, 1998. Springer.

[67] C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 314–328, Trento, Italy, 1999. Springer. To appear.

[68] C. Weidenbach, B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobalt, and D. Topic. System description: SPASS version 1.0.0. In H. Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 378–382, Trento, Italy, 1999. Springer. To appear.

[69] E. Weydert. Rational default quantifier logic. In D. Gabbay, R. Kruse, and A. Nonnengart, editors, *Proceedings of the 1st International Joint Conference on Qualitative and Quantitative Practical Reasoning (ESQARU-FAPR-97)*, volume 1244 of *Lecture Notes in Artificial Intelligence*, pages 589–600, Bad Honnef, Germany, June 1997. Springer.

[70] E. Weydert. System JZ : How to build a canonical ranking model of a default knowledge base. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pages 190–201, Trento, Italy, 1998. Morgan Kaufmann.

[71] J. Wu and M. Lu. CWA in multi-valued logics. In Z. Li, editor, *Proceedings of the 3rd Asian Symposium on Computer Mathematics (ASCM-98)*, pages 259–270, Lanzhou, China, August 1998. Lanzhou University, the Mathematics Mechanization Research Center (MMRC), and the Japanese Society for Symbolic and Algebraic Computation (JSSAC)., Lanzhou University.

Workshop proceedings

[1] A. Abdelwaheb, D. A. Basin, and A. Podelski. Lisa: A specification language based on ws2s. In M. Nielsen and W. Thomas, editors, *Proceedings of the 11th International Workshop on Computer Science Logic (CSL-97)*, volume 1414 of *Lecture Notes in Computer Science*, pages 18–34, Aarhus, Denmark, 1998. Springer.

[2] A. Bockmayr and Y. Dimopoulos. Mixed integer programming models for planning problems. In J. Frank and M. Sabin, editors, *Proceedings of the Workshop on Constraint Problem Reformulation (CP-98)*, pages 1–6, Pisa, Italy, 1998. NASA Ames Research Center.

[3] G. Delzanno. Specification of Term Rewriting in Linear Logic. In D. Galmiche, editor, *Proceedings of Workshop on Proof-Search in Type-Theoretic Languages*, volume 17 of *Eletrical Notes in Theoretical Computer Science*, Lindau, Germany, 1998. Elsevier Science.

[4] R. A. Schmidt. Relational grammars for knowledge representation. In M. Böttner, editor, *Proceedings of the Workshop on Variable-Free Semantics*, page 19, Osnabrück, Germany, 1999. Fachbereich Sprach- und Literaturwissenschaft, Univ. Osnabrück. To appear.

[5] V. Sofronie-Stokkermans. Representation theorems and automated theorem proving in certain classes of non-classical logics. In P. Eklund, G. Escalada-Imaz, R. Haehnle, and P. Vojtas, editors, *Proceedings of the Workshop on Many-Valued Logic for AI Applications (ECAI-98)*, Brighton, UK, August 1998. ECAI.

[6] J. Stuber. Strong symmetrization, semi-compatibility of normalized rewriting and first-order theorem proving. In M. P. Bonacina and U. Furbach, editors, *Proceedings of the International Workshop on First-Order Theorem Proving (FTP-97)*, volume 97-50 of *RISC-Linz Report*, pages 125–129, Schloss Hagenberg by Linz, Austria, 1997. Johannes Kepler Universität.

[7] L. W. N. van der Torre. Phased labeled logics of conditional goals. In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Proceedings of the 6th European Workshop on Logics in AI: Logics in Artificial Intelligence (JELIA-98)*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 92–106, Schloss Dagstuhl, 1998. Springer.

[8] L. W. N. van der Torre and Y.-H. Tan. Prohairetic deontic logic (pdl). In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Proceedings of the 6th European Workshop on Logics in AI: Logics in Artificial Intelligence (JELIA-98)*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 77–91, Schloss Dagstuhl, 1998. Springer.

[9] E. Weydert. Qualitative entropy maximization - A preliminary report. In E. Weydert, G. Brewka, and C. Witteveen, editors, *Proceedings of the Third Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR-97)*, pages 63–72, Saarbrücken, Germany, 1997. Max-Planck-Institut für Informatik.

[10] E. Weydert. Minimal information entailment : A preliminary account. In J. Delgrande and M. Truszczynski, editors, *Proceedings of the 7th International Workshop on Nonmonotonic Reasoning (Workshop on Formal Aspects and Applications of Nonmonotonic Reasoning)*, pages 64–72, Trento, Italy, 1998.

[11] E. Weydert. JZBR - Iterated belief change for conditional ranking constraints. In H. Rott, C. Albert, G. Brewka, and C. Witteveen, editors, *Proceedings of the Fourth Dutch-German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR 99)*, pages 57–66, Amsterdam, The Netherlands, 1999. ILLC Scientific Publications.

[12] E. Weydert and L. W. N. van der Torre. Goals, desires, utilities and preferences. In J. Lang, editor, *Proceedings of the ECAI-98 Workshop : Decision theory meets artificial intelligence - qualitative and quantitative approaches*, pages 57–64, Brighton, United Kingdom, 1998. ECAI.

Theses

[1] M. Christen. A calculus of simplification for superposition. Master's thesis, Universität des Saarlandes, 1997.

[2] S. Friedrich. Integration of a decision procedure for second-order monadic logic in a higher-order logic theorem proving environment. Master's thesis, Universität des Saarlandes, April 1998.

[3] T. Kasper. *A Unifying Logical Framework for Integer Linear Programming and Finite Domain Constraint Programming.* PhD thesis, Universität des Saarlandes, December 1998.

[4] P. Leven. Integrating clausal decision procedures in a tactic based theorem prover. Master's thesis, Universität des Saarlandes, 1998.

[5] R. Rau. Adaption of a fuzzy controller for a cybernetic system through an evolutionary algorithm. Master's thesis, Universität des Saarlandes, 1997.

[6] R. A. Schmidt. *Optimised Modal Translation and Resolution.* PhD thesis, Universität des Saarlandes, November 1997.

[7] G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic.* PhD thesis, Universität des Saarlandes, April 1999.

[8] J.-G. Timm. Testing the satisfiability of rpo constraints. Master's thesis, Universität des Saarlandes, 1997.

[9] L. Viganò. *A Framework for Non-Classical Logics.* PhD thesis, Universität des Saarlandes, 1997.

[10] U. Waldmann. *Cancellative Abelian Monoids in Refutational Theorem Proving.* PhD thesis, Universität des Saarlandes, July 1997.

Technical reports

[1] L. Bachmair and H. Ganzinger. Strict basic superposition and chaining. Research Report MPI-I-97-2-011, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, December 1997.

[2] L. Bachmair and H. Ganzinger. A theory of resolution. Research Report MPI-I-97-2-005, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, May 1997. To appear in J.A. Robinson, A. Voronkov (eds): Handbook of Automated Reasoning.

[3] L. Bachmair, H. Ganzinger, and A. Voronkov. Elimination of equality via transformation with ordering constraints. Research Report MPI-I-97-2-012, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, December 1997.

[4] D. A. Basin, S. Matthews, and L. Viganò. Labelled modal logics: quantifiers. Research Report MPI-I-97-2-001, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, January 1997.

[5] P. Blackburn and M. Tzakova. Two hybrid logics. Research Report MPI-I-97-2-007, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, July 1997.

[6] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. Research Report MPI-I-98-2-006, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, February 1998.

[7] A. Bockmayr and F. Eisenbrand. On the Chvátal rank of polytopes in the 0/1 cube. Research Report MPI-I-97-2-009, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, September 1997.

[8] A. Bockmayr and T. Kasper. A unifying framework for integer and finite domain constraint programming. Research Report MPI-I-97-2-008, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, August 1997.

[9] W. Charatonik. Automata on dag representations of finite trees. Research Report MPI-I-1999-2-001, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, March 1999.

[10] W. Charatonik and A. Podelski. Solving set constraints for greatest models. Research Report MPI-I-97-2-004, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, November 1997.

[11] V. Cortier, H. Ganzinger, F. Jacquemard, and M. Veanes. Decidable fragments of simultaneous rigid reachability. Research Report MPI-I-1999-2-004, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, March 1999.

[12] A. Degtyarev and A. Voronkov. Equality reasoning in sequent-based calculi. Research Report MPI-I-98-2-011, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, July 1998.

[13] G. Delzanno and A. Podelski. Model checking infinite-state systems in clp. Research Report MPI-I-98-2-012, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, July 1998.

[14] F. Eisenbrand. A note on the membership problem for the first elementary closure of a polyhedron. Research Report MPI-I-98-2-018, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, November 1998.

[15] H. Ganzinger, F. Jacquemard, and M. Veanes. Rigid reachability. Research Report MPI-I-98-2-013, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, September 1998.

[16] Y. Gurevich and M. Veanes. Partisan corroboration, and shifted pairing. Research Report MPI-I-98-2-014, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, October 1998.

[17] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logic. Research Report MPI-I-97-2-003, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, February 1997.

[18] F. Jacquemard, C. Meyer, and C. Weidenbach. Unification in extensions of shallow equational theories. Research Report MPI-I-98-2-002, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, December 1998.

[19] S. Ramangalahy. Strategies for conformance testing. Research Report MPI-I-98-2-010, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, May 1998.

[20] R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics. Research Report MPI-I-97-2-002, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, January 1997.

[21] R. A. Schmidt. E-unification for subsystems of s4. Research Report MPI-I-98-2-003, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, April 1998.

[22] V. Sofronie-Stokkermans. Resolution-based theorem proving for shn-logics. Technical Report E1852-GS-981, Technische Universität Wien, Vienna, Austria, November 1998. an extended version will appear in LNCS (subseries LNAI); Proceedings of FTP'98.

[23] M. Tzakova and P. Blackburn. Hybridizing concept languages. Research Report MPI-I-98-2-017, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, October 1998.

[24] M. Veanes. The relation between second-order unification and simultaneous rigid e-unification. Research Report MPI-I-98-2-005, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, February 1998. Extended version of a paper presented at LICS'98.

[25] S. Vorobyov. Third-order matching in $\lambda \rightarrow$-curry is undecidable. Research Report MPI-I-97-2-006, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, May 1997.

[26] S. Vorobyov. $\forall\exists^*$-equational theory of context unification is Π^0_1-hard. Research Report MPI-I-98-2-008, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, April 1998. Revised and abridged version appeared in the Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98), Springer Lecture Notes in Computer Science. Brno, Czech Republic, August 24-28.

[27] S. Vorobyov. The most nonelementary theory (a direct lower bound proof). Research Report MPI-I-98-2-007, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, April 1998.

[28] S. Vorobyov. Satisfiability of functional+record subtype constraints is NP-hard. Research Report MPI-I-98-2-004, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, 1998. An extended and revised version of this paper appeared in the Proceedings of the Annual Conference of the European Association for Computer Science Logic (CSL'98). Springer Lecture Notes in Computer Science.

[29] S. Vorobyov. The undecidability of the first-order theories of one step rewriting in linear canonical systems. Research Report MPI-I-98-2-009, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, May 1998.

[30] S. Vorobyov and A. Voronkov. Complexity of nonrecursive logic programs with complex values. Research Report MPI-I-97-2-010, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, November 1997.

[31] U. Waldmann. Cancellative superposition decides the theory of divisible torsion-free abelian groups. Research Report MPI-I-1999-2-003, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, March 1999.

[32] J. Wu. Symmetries in logic programs. Research Report MPI-I-1999-2-005, Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, May 1999.

Part V

Index

Index