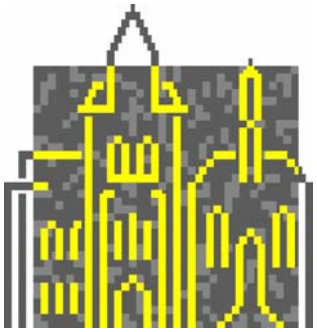

SCHLOSS DAGSTUHL

INTERNATIONAL
CONFERENCE
AND RESEARCH CENTER
FOR COMPUTER SCIENCE

Dagstuhl News

January - December 2001

Volume 4
2002

ISSN 1438-7581

Period: January - December 2001

Frequency: 1 per year

The International Conference and Research Center for Computer Science is operated by a non-profit organization. Its objective is to promote world-class research in computer science and to host research seminars which enable new ideas to be showcased, problems to be discussed and the course to be set for future development in this field.

Associates: Gesellschaft für Informatik e.V., Bonn
Technische Universität Darmstadt
Universität Frankfurt
Universität Kaiserslautern
Universität Karlsruhe
Universität Stuttgart
Universität Trier
Universität des Saarlandes

Funding: The state governments of Saarland and Rhineland Palatinate

Address: IBFI Schloß Dagstuhl
Octavieallee
D-66687 Wadern
Tel.: +49 - 6871 - 905127
Fax: +49 - 6871 - 905130
E-mail: service@dagstuhl.de
Internet: http://www.dagstuhl.de/

Welcome

You have in your hands the fourth edition of the "Dagstuhl News", a publication for the members of the Foundation "Informatikzentrum Schloss Dagstuhl", the *Dagstuhl Foundation* for short.

The main part of this leaflet consists of collected resumees and other hopefully interesting information excerpt from the Dagstuhl-Seminar Reports. We hope that you will find this information valuable for your own work or informative as to what colleagues in other research areas of Computer Science are doing. The full reports for 2001 are on the Web under URL: `http://www.dagstuhl.de/Seminars/01/`

Biggest news was the evaluation of our center by a committee of the German National Science Council (Wissenschaftsrat). As expected, the committee found, that, what we are doing, we were doing well. However, it felt that we had potential, which we could not exploit with the available resources. Hence, they asked us to sketch, how this additional potential could be tapped. This process is still going on. Another hearing is scheduled.

One of the foreseen changes will be that we would switch to publishing online proceedings of our Dagstuhl Seminars instead of the current Seminar Reports. Authors would keep the copyrights to their contributions in order not to harm their rights to submit them to conferences or journals. We hope that the reputation of our Dagstuhl Seminars will make their proceedings a valuable source of information.

The State and the Activities of the *Dagstuhl Foundation*

The foundation currently has 46 personal members and 8 institutional members. In 2001, the foundation has supported a few guests with travel grants and a reduction of the Seminar fees. According to German law only the interests earned can be used to support the aims of a foundation.

Thanks

I would like to thank you for supporting Dagstuhl through your membership in the *Dagstuhl Foundation*. Thanks go to Fritz Müller for editing the resumees collected in this volume.

Reinhard Wilhelm (Scientific Director)

Contents

1 Semantics in Databases

Seminar No. **01021** Report No. **295** Date **07.01.–12.01.2001**
Organizers: L. Bertossi, G.O.H. Katona, K.-D. Schewe, B. Thalheim

In the early days of database research, issues related to database semantics played a prominent role, and papers discussing database models, conceptual design, integrity constraints, normalization often dominated major database conferences. This began to change more than a decade ago, and nowadays those issues do not appear to be part of the mainstream database research. This situation is caused by several reasons: the problem began to be too difficult, the community was hoping on solutions based on better database models, the variety of buzzwords for new models and approaches required foundation and clarification, the problems raised by application required a lot of research, the pending hope on a universal, simple and genius solution. Nevertheless the semantical foundations are left open for most of the modern database models or are not existing for models such as UML or XML. At the same time, the community was forgetting the achievements of the early database research.

The seminar "Semantics in databases" will be a forum for researchers still interested in database semantics and which can contribute on the basis of research on database semantics and modern approaches of logics, algebra and combinatorics to the solution of very difficult problems raised in application. The first workshop "Semantics in Databases" has been organized in Rez in January 1995. The results of the discussions of the workshop have been summarized in the post workshop proceedings published in LNCS 1358.

Semantics of databases and information systems can be based on approaches which have been developed and successfully used by different communities: the logics community, especially those working on nonmonotonic reasoning, theorem proving, deduction, abduction, induction, finite model theory, constraint problems, non-classical semantics and categorical foundations; the type theory community, especially those working on algebraic foundations of information systems; the complexity theory community, especially those working on combinatorial foundations of information systems complexity; the database theory community which has been continuing research on constraints; the AI community, especially those working on agents, application of non-classical logics and

reasoning, deduction, abduction, induction in data and knowledge bases and non-monotonic reasoning; the database community, especially those contributing to foundations of information systems design including Web-based information systems, temporal aspects, integrity and security and database dynamics.

Research on semantics of databases is forced by a variety of problems challenging the modern information society. First, modern approaches are based on complex database and information system models. It requires the reformulation of classical results and their re-summarization including the restoration of older results. Dynamic semantics remains to be still open. Almost nothing is known on interaction semantics. Second, current technology faces difficult challenges. Information systems are becoming part of the everyday's infrastructure. They are used mainly by unexperienced users mainly on the basis of natural language understanding. The success of the Internet caused the utilization of huge varieties of semantics. XML and other ML are becoming the baseplate of Internet applications. However, XML supports semantics and consistency to a very limited extend. Third, the importance of research on semantics is hyper-raising by applications. Applications are more and more decentralized and require careful integration. UML became to be the 'inter-galactic' application specification language. It has almost no semantics. In order to develop consistent applications the semantics foundation needs to be worked out.

This Dagstuhl seminar aims in bringing together different communities such as the database theory community, the logics community, the AI community, the complexity theory community, the type theory community and the database applications community. The aim of the seminar is to provide a working environment where: different streams of semantics research are becoming aware of research of other communities; maintaining the database semantics community; working on plans for a survey on semantics in databases; discussing challenges of modern applications to semantics; figuring out research challenges of the next decade. To encourage discussion, we will have talks with a responder. The length of a talk is restricted to 60 minutes. The talk is followed by a discussion introduced by the responder of about 30 minutes overall.

Abstracts selected by the Dagstuhl News editor:

On No Information Nulls in Relational Databases

Hans-Joachim Klein, Kiel University, Germany

The interpretation of a missing data value as "the value is either unknown or not existent" is more difficult to handle correctly than the "unknown but existent" interpretation which has been studied quite extensively. Difficulties arise because the set of possible worlds described by a database with 'no information' nulls includes possible worlds with attribute values definitely known to not exist. Database semantics has to be defined carefully in order not to run into conflicts with the closed world assumption. To achieve this goal negative information has to be represented in a unique way. In general, additional information concerning relationships between attributes is necessary for guaranteeing unique representation.

In this talk we first demonstrate how semantics of the standard query language SQL allows to interpret null values by choosing appropriate query formulations. These formulations are equivalent for complete databases but not for databases with null values. Then we discuss sure and maybe information in answers and show why many proposals for query evaluation given in the literature do not guarantee sure information. In order to cope with problems in connection with the closed world assumption and not-existent values in possible worlds we introduce so-called concepts. For databases with concepts given for relation types in the corresponding schema, we define semantics based on extensions of relations. Then we show how to evaluate queries efficiently by applying a "switch strategy" such that the answers always represent sure information w.r.t. this semantics. Because of well-known efficiency problems, answers are not complete in general. The method can be applied to the standard query language SQL where missing values are modeled by "no information" nulls.

Paraconsistency: A Non-Standard Perspective of Logic, Resolution and Database Semantics

Hendrik Decker, Siemens Business Services, Germany

The purpose of this talk is to raise curiosity about paraconsistency among the participants of the seminar. Its ultimate goal is to convince them that paraconsistency should really be the standard logic foundation for the semantics of databases.

Classical logic takes an absolutely intolerant stand on inconsistency, although the latter is commonplace in reality. In particular, classical logic fails to capture that, in practice, querying inconsistent deductive systems ordinarily produces reasonably useful answers. Paraconsistent logic restricts classical logic so that, from inconsistency, not everything becomes derivable. We briefly survey some formal aspects of the history of mathematical logic, from Aristotle via Frege, Russel, Brower, Heyting, Kolmogorov, Lukasiewicz to Jasowski and da Costa. In this survey, we focus on axioms and inference rules used or renounced in classical logic, intuitionism, paraconsistency, resolution, logic programming, abduction and deductive databases. We argue that (even without extensions by annotation, additional truth values, probabilistic or other constructs), resolution, logic programming, abductive logic programming and deductive databases already have the potential of capturing a practically viable kind of paraconsistency.

2 Interoperability of Reverse Engineering Tools

Seminar No. **01041** Report No. **296** Date **21.01.–26.01.2001**
Organizers: Jürgen Ebert, Kostas Kontogiannis, John Mylopoulus

Software Reengineering is the present-day term for all activities for renovating aging systems to be more responsive to changes. Problems of the 90s like the Y2k-problem or the problem of converting software to the new European currency witnessed the importance of concepts, tools and techniques to improve the quality and maintainability of software.

Reengineering is a part of software engineering with its focus on all problems appearing during software maintenance of legacy software. In

this seminar we focused on the technical part related to the software artifacts themselves and excluded the management aspects of software maintenance.

Reengineering activities use to a large part the same concepts, tools and techniques as other software engineering disciplines. Besides software engineering knowledge, there is also much usage of other more traditional areas of computer science, especially compiler construction, database systems, formal semantics, and knowledge representation. But due to its special focus, additional problems appear, like

- reverse engineering
  (recognition of architecture, cliches, procedures, structure, and redocumentation),
- migration
  (change of database models and/or programming languages), and
- program understanding
  (querying, browsing, visualization techniques, concept analysis).

Software Reengineering activities are widespread and mostly focus on the development of tools for software analysis. Thanks to several conferences and workshops in this area such as CSMR, ICSM, IWPC and WCRE a common terminology and definitions of agreed aims of reengineering research are slowly emanating.

The current state of practice is that reengineering tools still solve insular problems and are treated as research prototypes within the research group that developed them. In this context, it is very important to define a data interchange format that allows for different reengineering tools to communicate so that integrated, multi-faceted representations of software systems can be created.

Even though this issue may look simple at the beginning, it involves a number of research issues to be resolved. One issue is the definition of the levels of abstraction that information about a software system is to be presented on. These levels of abstraction may include the abstract syntax tree level, the data and control flow level, or the architectural level.

The challenge to the research community is to design a formalism for each level of abstraction so that information about a software system can

be passed from one analysis tool to another. Moreover, the formalism must allow for software systems and constructs in various languages to be presented. Another research challenge is the definition of schemas that allow for data emitted from different parsers to be fused in a uniform, normalized source code representation. Emerging markup languages such as XML may provide a vehicle for data fusion and data integration in this context.

The actions done for enhancing interoperability of research and tools in reengineering have only partially been successful up to now. Schemas for a number of popular languages such as C, Cobol, Fortran and, PL/I have been developed. Languages and supporting tools that allow for architectural descriptions to be specified and exchanged in the form of tuples have been implemented by various groups.

At this 5-day Dagstuhl seminar 47 people gathered together in order to discuss, extend and combine the work done by various groups into a common confluent and coherent result. This report collects the abstracts of all talks given during this week covering all relevant aspects of interoperability - including metamodels, concrete tools and frameworks, practical experiences and relevant technologies like XML. One afternoon was used for a demo session where several tools were introduced to the participants (db-main, shrimp, codecrawler, gupro, columbus, ta, missinglink, fujaba).

The participants agreed on the necessity of interchange languages on different levels of semantic expressiveness, like

- abstract syntax trees and abstract syntax graphs,
- call graphs and program dependence graphs,
- architecture descriptions.

XML was regarded as a vehicle for reengineering tool interoperability. GXL, a proposal for an XML-based basic format for interchanging graph-like reengineering data, had been developed during the last months before the seminar by the cooperation of several groups, especially Waterloo (Ric Holt), Toronto (Susan Elliott Sim), Koblenz (Andreas Winter) and Munich (Andy Schürr). GXL is an XML-based amalgamation of existing formats. On friday, January 26th, the participants

of this seminar accepted GXL 1.0 (meanwhile known as the Dagstuhl-version) as an interchange language and more than twenty participants assured that they were going to make their tools interoperable on the basis of GXL. The further development of GXL can be inspected at `http://www.gupro.de/GXL/`.

Beyond a common format, tool interoperability highly relies on the agreement of common concepts, schemas, and information structures. During the workshop the participants discussed such schemas in four groups focussing on different aspects: syntax level, middle level, architecture level and data level. The discussions started by these groups did not yet lead to mature proposals, but this work is still being continued. Concerning the C++ syntactical schema a lively email-list (`gxl-cpp@rgai.inf.u-szeged.hu`) was founded where the discussion is continued. Concerning the middle level a wiki has been installed at `http://scgwiki.iam.unibe.ch:8080/Exchange/2`.

So this seminar was not only a productive and pleasant week for all participants, largely thanks to the well thought-out setup and the quality of service that Dagstuhl provides. It also was the source of continuing work on interoperability of reengineering tools.

3 Computer Aided Design and Test: BDDs versus SAT

Seminar No. **01051** Report No. **297** Date **28.01.–02.02.2001**
Organizers: Bernd Becker, Masahiro Fujita, Christoph Meinel, Fabio Somenzi

The focus of the sixth workshop in the biannual series *Computer Aided Design and Test* at the IBFI Schloß Dagstuhl was on **BDDs vs. SAT**. The seminar was organized by Bernd Becker (University Freiburg), Masahiro Fujita (University of Tokyo), Christoph Meinel (University Trier), and Fabio Somenzi (University of Colorado). It was attended by 44 scientists.

While after 10 years use of BDDs various BDD-based algorithms have been developed and BDD-techniques have seen dramatic improvements only recently, SAT based techniques are reconsidered with respect to their usability in Electronic Design Automation and in other applications.

The organizers took the opportunity to bring together researchers from different areas in computer science, electrical engineering and industry. During the seminar 31 lectures covering different aspects of the topic were presented and the seminar provided a forum for scientific discussion e.g. on

- both approaches, also on comparisons among various approaches to SAT,
- the advances in BDD and SAT algorithms,
- comparisons between BDDs and SAT for various applications e.g., model checking,
- hybrid approaches that use BDDs and SAT, and
- other approaches to the decision of Boolean formulae.

More detailed information including some full papers can be found on the WWW-pages with the URL:

- `http://www.dagstuhl.de/DATA/Seminars/01/`
- `http://www.bdd-portal.org/dagstuhl-ppt/dagstuhl-talks.htm`

Abstracts selected by the Dagstuhl News editor:

Improved OBDD and FBDD Lower Bounds for Integer Multiplication via Universal Hashing

Beate Bollig, University Dortmund, Dortmund, Germany, joint work with Philipp Woelfel

Binary Decision Diagrams (BDDs) are graph representations for Boolean functions. Besides the complexity theoretical viewpoint people have used restricted BDDs in applications where the complexity of fundamental functions is of interest. FBDDs are BDDs where on each path from the source to a sink each variable is tested at most once. OBDDs, one of the most popular representations in applications, have the additional restriction that on all paths the variables are tested according to a given variable ordering.

Bryant (1991) has shown that any OBDD representation for the function $\mathrm{MULT}_{n-1,n}$, which computes the middle bit of the product of two n-bit numbers, requires at least $2^{n/8}$ nodes. This bound would still allow the possibility that one can construct 64-bit multipliers represented by OBDDs containing only 256 nodes, where on the other hand it is widely conjectured that OBDDs computing $\mathrm{MULT}_{n-1,n}$ have a size of at least 2^n. In this talk a stronger lower bound of $\frac{1}{61}2^{n/2}$ is proven by a new technique using a recently found universal family of hash functions.

Ponzio (1995, 1998) has presented a lower bound of $2^{\Omega(n^{1/2})}$ on the size of FBDDs for $\mathrm{MULT}_{n-1,n}$. Combining results and methods for universal hashing with lower bound techniques for FBDDs the first strongly exponential lower bound of $\Omega(2^{n/4})$ is proven for the middle bit of integer multiplication.

Dynamic Selection of Branching Rules

Marc Herbstritt, Albert-Ludwigs-University, Freiburg, Germany

Current SAT solvers (e.g. GRASP) consist of three "engines": the deduction engine, the diagnosis engine, and the decision engine. Branching rules are applied in the decision engine to select a variable and an assignment to this variable to guide the search process. In the last years several branching rules were developed, but there is no "best-of-all" branching rule. Another powerful technique to speed up search is non-chronological backtracking which is part of the diagnosis engine. Due to non-chronological backtracking it can be avoided to search "senseless" parts of the search tree.

In this talk we present a method to combine information from non-chronological backtracking and the pool of available branching rules. The intuition behind our approach is that the branching rule which caused a conflict and thus led to a backtrack should be "punished". Therefore we maintain preference values for all branching rules which model the probability to be selected when a decision assignment is made. To punish a branching rule we count how often it was used and how often it triggered a conflict. These values are used to diminish the preference value of the branching rule. To select a branching rule during decision assignment we use well known selection methods (roulette-wheel, linear ranking, tournament selection).

Our approach results in a faster and more robust behaviour of the SAT solver.

4 Applications of Kleene Algebra

Seminar No. **01081** Report No. **298** Date **18.02.–23.02.2001**
Organizers: Roland Backhouse, Dexter Kozen, Bernhard Möller

Kleene algebra (KA) [23, 16, 24] is an algebraic system for calculating with sequential composition, choice and finite iteration. It was first introduced by Kleene in 1956 and further developed by Conway in 1971. It has reappeared in many contexts in mathematics and computer science. Its classical application has been within the theory of formal languages, where it is one of many equivalent approaches to the description of regular languages.

Within the field of efficient algorithms it has been applied to path problems on graphs (being closely related to the algebra of closed semirings [2]), to convex hull algorithms and formal treatment of pointer algorithms [5, 6, 10, 14, 12].

In compiler construction, Kleene algebra can be used to prove the correctness of optimization techniques for loop constructs [25].

More recently, Kleene algebra has been successfully applied to the semantic description of imperative programs with non-deterministic choice [4]. It covers both angelic and demonic composition and choice [17, 18].

Moreover, it allows a simple algebraic incorporation of assertions [21] as well as modal and dynamic logic [31, 26]. Also, there are close relations with interval and temporal logic [29, 22], the duration calculus [33, 34] and timed automata [3]. Further applications concern switching theory [30].

Finally, a particular form of Kleene algebra corresponds to quantales with unit elements [32, 1, 11]. Hence there is a correspondence with linear logic.

Extensions of Kleene algebra deal with infinite iteration. They have been used in the description and verification of protocols and in proofs about concurrent systems in general [15]. Related systems are iteration theories [9], the computation calculus [19] and the theory of ω-languages. Dropping one of the distributivity requirements for Kleene algebras leads

to a system that is close to process algebras such as ACP or μCRL [7, 20]. Finally, there are close connections to network algebra [8, 13].

These tracks of research have so far been undertaken in a rather isolated manner. The aim of this seminar was to bring the researchers from these tracks together for fruitful interaction and for helping the subject to more public visibility. To our knowledge, this was the first international sysmposion dedicated to the *applications* of Kleene algebra.

Compared with other algebraic approaches to semantics, such as relation algebra or sequential calculus, Kleene algebra and its relatives enjoy a particularly simple axiomatisation, since they do not use a notion of (pseudo-)converse and the corresponding axioms.

Since its basic features and rules are known even to beginner students of computer science, Kleene algebra will be more easily accepted than other formal systems and thus may serve as an effective vehicle for formal treatment of various subjects in computer science.

Abstracts selected by the Dagstuhl News editor:

Factor Theory Revisited

Roland Backhouse, University of Nottingham

Conway's book "Regular algebra and finite machines" has been mentioned frequently at this seminar. However, one important contribution in his book that has not been mentioned is his study of so-called "factors" of a regular language. The goal of this talk is to bring this to everyone's attention. Conway's "factors" are called "residuals" in relation algebra and "weakest prespecifications" in the programming literature; the fact that the same concept is known by various names attests to its importance. Conway's contribution was to show that the factors of a regular language can be organised in a matrix, which he called the factor matrix. This matrix has a number of special properties – for example, the matrix is reflexive and transitive. Conway's account of the factor matrix is however very wordy, making it difficult to read and check. In one case, the unfortunate omission of the word "not" in a sentence caused me a great deal of confusion when first reading his text!! In this talk, I show how factors are formulated using the now standard Galois connection defining the residuals of a relation. This allows one to give precise, calculational formulations of the factor matrix. I also mention the relation between

the factor "graph" and the Knuth, Morris, Pratt string matching algorithm (see Backhouse and Lutz, ICALP 1977). Prompted by an earlier talk I show how factors are used to formulate the well-foundedness of a relation. This formulation is used to present a calculational proof of Newman's lemma (see Doornbos, Backhouse and Van der Woude, TCS, 1997).

Kleene-ing Up Semantics

Bernhard Möller, Universität Augsburg, partially joint work with Jules Desharnais

Kleene algebras provide a convenient and powerful algebraic axiomatisation of a complete lattice that is endowed with a sequential composition operation. The particular kind of Kleene algebras we are considering is equivalent to Boolean quantales. Models include formal languages under concatenation, relations under standard composition, sets of graph paths under path concatenation and sets of streams under concatenation.

The least and greatest fixpoint operators of a complete lattice allow definitions of the finite and infinite iteration operators * and $^\omega$, resp.

Elements of Kleene algebras can be used, among others, as abstractions of the input-output semantics of nondeterministic programs or as models for the association of pointers with their target objects. In the first case, one seeks to distinguish the subclass of elements that correspond to deterministic programs. In the second case one is only interested in functional correspondences, since it does not make sense for a pointer to point to two different objects.

We discuss several candidate notions of determinacy and clarify their relationship. Some characterizations that are equivalent in the case where the underlying Kleene algebra is an (abstract) relation algebra are not equivalent for general Kleene algebras.

In relational semantics, the input-output semantics of a program is a relation on its set of states. We generalize this in considering elements of Kleene algebras as semantical values. In a nondeterministic context, the demonic semantics is calculated by considering the worst behavior of the program. In this paper, we concentrate on while loops. While calculating the semantics of a loop is difficult, showing the correctness of any candidate abstraction is much easier. For deterministic programs, Mills

has described a checking method known as the while statement verification rule. A corresponding programming theorem for nondeterministic iterative constructs is proposed, proved and applied to an example. This theorem can be considered as a generalization of the while statement verification rule to nondeterministic loops.

In standard Kleene algebra it is assumed that the composition operation is universally disjunctive in both arguments. This entails monotonicity and strictness w.r.t. the least element 0 that plays the role of $\perp$ in denotational semantics. However, full strictness does not make sense when one wants to give an algebraic account of systems with lazy evaluation. Therefore we study a "one-sided" variant of KAs in which composition is strict in one argument only. This treatment fits well with systems such as the calculus of finite and infinite streams which is also used in R. Dijkstra's computation calculus.

There is some choice in what to postulate for the other argument. Whereas Dijkstra stipulates positive disjunctivity, we investigate how far one gets if only monotonicity is required. The reason is that we want to enable a connection to process algebra. There only one of the distributivity laws for composition over choice is postulated to preserve the temporal succession of choices.

References

[1] S. Abramsky, S. Vickers: *Quantales, observational logic and process semantics.* Math. Struct. Comp. Science **3**, 161–227 (1993)

[2] A.V. Aho, J.E. Hopcroft, J.D. Ullman: *The design and analysis of computer algorithms.* Reading, Mass.: Addison Wesley 1974

[3] E. Asarin, O. Maler, P. Caspi: *A Kleene theorem for timed automata.* In: G. Winskel (Ed.): Proc. LICS'97, 160-171, 1997.

[4] R.-J. Back, J. von Wright: *Reasoning algebraically about loops.* Acta Informatica **36** 295–334 (1999)

[5] R.C. Backhouse, B.A. Carré: *Regular algebra applied to path-finding problems.* J. Institute of Mathematics and its applications **15**, 161–186 (1975)

[6] R.C. Backhouse, A.J.M. van Gasteren: *Calculating a path algorithm.* In: R.S. Bird, C.C. Morgan, J.C.P. Woodcock (eds.): Mathematics of program construction. Lecture Notes in Computer Science **669**. Berlin: Springer 1993,32–44

[7] J.A. Bergstra, I. Bethke, and A. Ponse: *Process algebra with iteration and nesting.* The Computer Journal, **37**, 243-258 (1994)

[8] J.A. Bergstra, G. Ştefănescu: *Network algebra with demonic relation operators.* Report P9509, PRG, University of Amsterdam, 1995. Revised Version: Revue Roumaine de Mathematique Pure et Appliquée (to appear)

[9] S.L. Bloom, Z. Esik: *Iteration Theories: The Equational Logic of Iterative Processes.* Berlin: Springer 1993

[10] Francis Bossut, Max Dauchet, and Bruno Warin: *A Kleene theorem for a class of planar acyclic graphs.* Information and Computation, **117**, 251–265 (1995).

[11] C. Brown, D. Gurr: *A representation theorem for quantales.* Journal of Pure and Applied Algebra, 85:27-42 (1993)

[12] T. Brunn, B. Möller, M. Russling: *Layered graph traversals and hamiltonian path problems – An algebraic approach.* In: J. Jeuring (ed.): Mathematics of Program Construction. Lecture Notes in Computer Science **1422**. Berlin: Springer 1998, 96–121

[13] V.E. Cazanescu, G. Ştefănescu: *Feedback, iteration and repetition.* INCREST Preprint 42/1988. Published in: G. Paun (ed.): Mathematical aspects of natural and formal languages. World Scientific 1994, 43-62.

[14] K. Clenaghan: *Calculational graph algorithmics: reconciling two approaches with dynamic algebra.* CWI Amsterdam, Report CS-R9518, 1995

[15] E. Cohen: *Separation and reduction.* In: R. Backhouse, J. Oliveira (eds.): *Mathematics of program construction.* Lecture Notes in Computer Science **1837**. Berlin: Springer 2000, 45–59

[16] J.H. Conway: *Regular algebra and finite machines.* London: Chapman and Hall 1971

[17] J. Desharnais, B. Möller: *Characterizing determinacy in Kleene algebra.* Special Issue on Relational Methods in Computer Science, Information Sciences — An International Journal (to appear)

[18] J. Desharnais, B. Möller, F. Tchier: *Kleene under a demonic star.* In: T. Rus (ed.): *Algebraic Methodology and Software Technology.* Lecture Notes in Computer Science **1816**. Berlin: Springer 2000, 355–370

[19] R.M. Dijkstra: *Computation calculus bridging a formalization gap.* Science of Computer Programming **37**, 3–36 (2000)

[20] W.J. Fokkink: *Axiomatisations for the perpetual loop in process algebra.* In: P. Degano, R. Gorrieri, A. Marchetti-Spaccamela (eds.): Proc. 24th ICALP. Lecture Notes in Computer Science **1256**. Berlin: Springer 1997, 571–581

[21] M. Hollenberg: *Equational axioms of test algebra.* In: M. Nielsen, W. Thomas (Eds.): Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL, Aarhus, Denmark, August 23-29, 1997, Selected Papers. Lecture Notes in Computer Science **1414**. Berlin: Springer 1998, 295-310

[22] B. von Karger: *Temporal algebra.* Universität Kiel, Habilitationsschrift 1997

[23] S.C. Kleene: *Representation of events in nerve nets and finite automata.* In: C. Shannon, J. McCarthy (eds.): Automata Studies. Princeton University Press 1956, 3–41

[24] D. Kozen: *A completeness theorem for Kleene algebras and the algebra of regular events.* Information and Computation **110**, 366-390 (1994)

[25] M.-C. Patron, D. Kozen: *Certification of compiler optimizations using Kleene algebra with tests*, Report 99-1779, Computer Science Department, Cornell University, Dec. 1999.

[26] D. Kozen, J. Tiuryn: *On the completeness of propositional Hoare logic.* In: J. Desharnais (ed.): *RelMiCS 2000, 5th International Seminar on Relational Methods in Computer Science.* Université Laval, Québec, Jan. 2000, 195-202.

[27] B. Möller: Towards pointer algebra. Science of Computer Programming **21**, 57–90 (1993)

[28] B. Möller: Calculating with acyclic and cyclic lists. Special Issue on Relational Methods in Computer Science, Information Sciences — An International Journal **119/3–4**, 135–154 (1999)

[29] B. Moszkowski: Some very compositional temporal properties. In: E.-R. Olderog (ed.): Programming concepts, methods and calculi. IFIP Transactions A-56. Amsterdam: North-Holland 1994, 307–326

[30] T. Ninomiya, M. Mukaidono: *Clarifying the axioms of Kleene Algebra based on the method of indeterminate coefficients.* Proc. 29th IEEE International Symposium on Multiple-Valued Logic (ISMVL 99), Albert-Ludwigs-University, Freiburg im Breisgau , Germany, on May 20-22, 1999

[31] V. Pratt: *Dynamic algebras as a well-behaved fragment of relation algebras.* In: C.H. Bergman, R.D. Maddux, D.L. Pigozzi (eds.): Algebraic Logic and Universal Algebra in Computer Science. Lecture Notes in Computer Science **425**. Berlin: Springer 1988, 77–110

[32] K.I. Rosenthal: *Quantales and their applications.* Pitman Research Notes in Mathematics Series, Vol. 234. Longman Scientific & Technical 1990

[33] Zhou, C.: *Duration calculi: an overview.* In: D. Bjørner, M. Broy, I.V. Pottosin (eds.): Formal methods in Programming and their Applications. Lecture Notes in Computer Science **735**. Berlin: Springer 1993, 256–266

[34] Zhou, C., A.P. Ravn, H. Rischel, J.U. Skakkebæk: *Specification of embedded, real-time systems.* Proc. 1992 Euromicro Workshop on Real-Time Systems. IEEE Computer Society Press 1992, 116-121

5 Algorithmic Techniques in Physics

Seminar No. **01091** Report No. **299** Date **25.02.–02.03.2001**
Organizers: Michael Jünger, Gerhard Reinelt, Heiko Rieger, Giovanni Rinaldi

Nearly three years earlier, in December 1997, the Dagstuhl Seminar "Algorithmic Techniques in Physics" took place. Researchers from Computer Science, Mathematics and Physics came together to discuss about algorithmic problems occurring in physics and physical concepts that might be useful in computer science. Bringing together people from three different areas was an experiment that, as all participants agreed in the end, turned out to be a success and, more importantly, should be repeated in the future.

The original seminar was motivated by the observation that traditionally, there has always been a strong scientific interaction between physicists and mathematicians in developing physical theories. However, even though numerical computations are now commonplace in physics, no comparable interaction between physicists and computer scientists had been developed. Since the last three decades the design and the analysis of algorithms for decision and optimization problems evolved rapidly. Simultaneously, computational methods in theoretical physics became a major research tool causing a fast growing challenge with regards to the underlying algorithmic concepts. The few interactions between physicists and computer scientists were often successful and provided new insights in both fields. For example, in one direction, the algorithmic community has profited from the introduction of general purpose optimization tools like the simulated annealing technique that originated in the physics community. In the opposite direction, algorithms in linear, nonlinear, and discrete optimization have turned out to be useful tools in physics. Surprisingly often physicists and computer scientists are concerned with very similar questions but use a different terminology disguising in this way the significant overlap and preventing fruitful collaboration. Many notions of physicists in particle physics the computer scientists call problems or algorithms in combinatorics and extremal graph theory.

During the first seminar it became clear that the communication that was intended by the organizers was indeed fruitful. The computer scientists realized rather quickly that computational physicists often deal with very similar problems and try to solve them with a sometimes more

pragmatic approach. And the physicists profited from the most recent algorithmic developments that were useful for them but usually reach their community only decades later. Actually a number of participants (including the organizers) took home various ideas that were later converted into scientific publications. Various algorithms from combinatorial optimization are now standard tools in the computational physics community that studies the properties of ground states of disordered and complex systems. Some problems have been solved, but new ones have emerged. On the other hand, new algorithmic techniques have been developed which might be useful in solving these new problems. The topics that we treated in the second seminar include:

- Applications of polynomial optimization algorithms, including their problem specific implementation, to random physical systems. Well established examples here are the max-flow/min-cut algorithm to disordered ferromagnets or the min-cost-flow-algorithm to ensembles of magnetic flux lines.
- New developments in exact and heuristic algorithms for NP-hard physical problems, like the Coulomb- or the spin glass as well as new approaches to stochastic optimization. This included simulated annealing, genetic algorithms, semidefinite programming, Lagrangean relaxation, branch and cut, the Pfaffian method etc.
- Results on the nature of disordered systems and networks
- Concrete physical realizations of various standard problems in combinatorial optimization.
- The computational complexity of various computationally hard physical problems as they occur in the physics of glassy systems.
- Statistical properties and phase transitions of various standard problems in computer science like k-Sat etc.
- Scaling behavior of various geometric properties of standard algorithms applied to grid graphs.
- And many more.

We found that since the last seminar many language barriers between physicists, mathematicians and computer scientists have been overcome.

In a relaxed research atmosphere, the proposed problems were modeled and analyzed in consistent terminology understandable to all three groups. Therefore, an intensive information exchange was possible. This improvement became particularly apparent in the open forum discussions, where everybody was welcome to propose a problem and ask for possible solution methods.

Interestingly, it became clear that there is an increasing overlap in the research fields of both communities: physicists become more and more interested in combinatorial optimization problems and study their physical nature, whereas computer scientists realize the importance of studying physically relevant problems.

Our goal was to bring together the computer science, mathematical programming and physics communities in the pursuit of establishing new interactions and refreshing old ones, initiated in the '97 event. It turned out that we had chosen a good time for the second such Dagstuhl seminar.

6 Methodology of Evaluation in Medical Image Computing

Seminar No. **01111** Report No. **301** Date **11.03.–16.03.2001**
Organizers: Kevin W. Bowyer, Murray H. Loew, H. Siegfried Stiehl, Max A. Viergever

About one decade ago, Yannis Aloimonos complained that "Unfortunately, there is a disconcerting lack of visual systems which perform well in real-world environments, particularly when compared to the amount of mathematical theory published on the subject."—a complaint which not only holds for computational vision in general but in particular also for the safety-critical case of medical image computing (a terminus technicus which commonly subsumes medical image formation, processing, analysis, interpretation, and visualization). One reason for this unfortunate situation is clearly the fact that the experimental basis of computational vision as a scientific discipline is still rather weak. As a down-to-earth-consequence, e.g., it is by no means clear for an industrial system designer, on which grounds she/he should rely on a particular algorithm, method, or proposed tool once she/he is faced with the problem of putting

academic research to work. Neither it seems to be clear for a clinician, what kind of as well as what degree or quality of support in his routine work she/he can expect from proffered medical image computing (MIC) tools claimed to support routine work. Put in other words, MIC seen as a coin has a shiny and scientifically rewarding theory side but a rather rusty, not to say puny, practice side.

Meanwhile in the MIC community a growing awareness of the fact can be observed that evaluation aiming at performance characterization is a critical issue. In a complementing way, a strong need from both clinical and industrial actors for tackling theoretical as well as experimental problems associated with this issue has to be stated, since dissemination of theoretical advances into practical settings requires a deep understanding of assets, limitations, application scope, etc. of MIC algorithms, methods, and tools. Moreover it is safe to state that without such a deep understanding gained from a scientific approach the design of interactive MIC systems will be severely hampered, since human-centered efficient interaction should take place on the basis of results of computational processes which are trustworthy—ideally results, which are consistent with theoretical proofs of a computational theory. In contrast to other application domains drawing upon visual data, MIC stands out for reasons of required safety, accuracy, robustness, ergonomy, etc. Apart from that, MIC is seen as a major future high-tech market also, hence the development of successful products strongly depends on bridging the gap between theory, experiment, and practice. Obviously, solutions to these problems reside in a space composed of multiple dimensions to name a few: MIC theory, practice of MIC (incl. design of algorithms and visual data structures), clinical requirement analysis, and industrial platform constraints.

Due to the lack of well-grounded, internationally accepted, and standardized methods for evaluation and given the specifity of MIC as briefly sketched above, it was high-time to bring together leading experts from the MIC community in the inspiring atmosphere of Schloss Dagstuhl to discuss the state-of-the-art/technology as well as routes to be jointly taken in the near future. After the successful first seminar on more general issues of performance characterization in computational vision in 1998 and given the most recent publications on domain-unspecific topics of evaluation in computational vision (see, e.g., “Empirical Evaluation Techniques in Computer Vision”, edited by K.W. Bowyer and J. Phillips, IEEE Computer Society Press, 1998; “Proc. Workshop on Per-

formance Characterisation and Benchmarking of Vision Systems (Las Palmas de Gran Canaria, Canary Islands, January 1999)", edited by A. Clark and P. Courtney; IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, No. 4, April 1999, Special Section on "Empirical Evaluation of Computer Vision Algorithms", edited by P.J. Phillips and K.W. Bowyer; "Performance Characterization and Evaluation of Computer Vision Algorithms", edited by R. Klette, H.S. Stiehl, M.A. Viergever, and K.L. Vincken, Kluwer Academic Publishers, 2000; "Tutorial on Performance Characterisation of Computer Vision Techniques (European Conf. on Computer Vision, Dublin, Ireland, June 2000)" by P. Courtney and N. Tacker; "Proc. 2. Workshop on Empirical Evaluation Methods in Computer Vision (Dublin, Ireland, June 2000)", edited by H.I. Christensen and P.J. Phillips, published as CPAV technical report no. 243, Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, June 2000), this seminar will focus on particular domain-specific issues as related to medical imagery, e.g. performance characterization of computational processes for segmentation, analysis, registration, and real-time visualization of multi-dimensional and multi-modal images.

In terms of priority, the focus w.r.t. presentations and discussions has been set on the following concrete topics:

1. validation and evaluation of accuracy, robustness, etc. of algorithms for interactive/semi-automatic/automatic segmentation, analysis, registration, and visualization of medical imagery

2. theoretical/methodological issues such as definition of ground truth and gold standards, value of phantoms, imaging simulators, and synthetic test data

3. selection of a representative set of clinical routine images related to specific domains and tasks (certified clinical reference cases and test image data base)

4. identification of open questions (see appendix)

One of the main goals of the successful seminar was to contribute towards a more seamless methodology of validation, evaluation, and performance characterization across various levels—thus to contribute also to bridge the gap between MIC theory and the end user and to provoke

fruitful discussions beyond the ivory tower. Despite the recent progress and achievements reported by the seminarians, it became quite clear during the week we had at our disposal that the MIC community has to retain the seminar topic on its research agenda for the years to come.

Throughout the week the moderators of the different sessions were asked by the organizers to collect hot topics and crucial open problems (coined "burning burning questions" or, as acronym, BBQs) in written form on a flip-chart. The final list was made up of the following BBQs (in random order):

1. characterization and formalization of noise in imagery
2. modelling of shading effects in e.g. microscopy or MR images
3. effect of preprocessing (e.g. image protocol selection, noise reduction, shading correction, contrast enhancement) on subsequent processing stages
4. influence of pathology in images on design of computational processes
5. importance of volumetric nature of image data
6. impact of anisotropic (e.g. CT and MR) image data on design of computational processes
7. effectiveness of computer graphics visualization for validation purposes
8. importance of unique terminology of validation and evaluation
9. relation of validation and evaluation through visual assessment to characteristics of visual system
10. mathematical foundation of image registration metrics
11. importance of correspondence problem in image registration
12. analysis of effect of local and global shape on computational processes
13. analysis of discretization (sampling and quantization) effects given continuous theory of computational processes

14. importance of spatial scale(s) for the design of computational processes
15. appropriateness of morphing (e.g. elastic registration) approaches w.r.t. anatomical variability
16. availability of ground truth
17. availability of annotated test image databases
18. importance of certification and of sharing of code
19. necessity of revealing assumptions of algorithms

7 Computational Geometry

Seminar No. **01121** Report No. **302** Date **18.03.–23.03.2001**
Organizers: Rolf Klein, Günter Rote

51 participants attended this meeting. As in previous Dagstuhl seminars about Computational Geometry, the exchange of ideas and information about the latest developments in the fields of computational and combinatorial geometry was the main purpose of this meeting. In addition, there was a focus on a few topics on individual days of the meeting, featuring longer keynote talks as well as more technical contributions:

- Geometric shape recognition, shape reconstruction and shape matching (talks 9, 10, 14, 26, 31, 33)
- Surface and volume modeling; optimal triangulations and meshing (2, 6, 7, 25, 29, 34, 35); visualization and rendering (1, 2, 11, 32)
- Robust and exact geometric computations (16, 17, 18)

We had originally planned to have also a focus on geometric methods in bioinformatics, exploring how and to what extent geometry-based methods can play a role in certain computational problems of biology such as molecular modeling, protein folding, drug design etc. However, we could not attract the bioinformatics experts to come to a week-long seminar on a field which is not central to their study, because it seems

that people who are active in bioinformatics are too busy in their own field.

In the core area of computational geometry, the contributions can be classified according to their subjects and techniques as follows.

- proximity and covering (1, 5, 8, 12, 19, 20)
- combinatorial geometry (3, 4, 13, 14, 15, 24, 27)
- dynamic geometry (3, 21, 22, 23, 25)
- data structures (28, 30)

8 Semantic Foundations of Proof-search

Seminar No. **01141** Report No. **303** Date **01.04.–06.04.2001**
Organizers: David Pym, Eike Ritter, Thomas Streicher

Description

Traditionally, logics are formulated as systems of deductive inference in which proofs are constructions which derive conclusions from given assumptions. However, in computing, many problems are naturally formulated as questions of reductive inference in which the correctness of a given putative conclusion must be shown by reduction, commonly formulated as proof-search in a given formal system, to established acceptable assumptions. Examples of this phenomenon include type-inference, parsing, program correctness and internet information retrieval. Typically, such examples are described as long and complex formal texts. Consequently, algorithmic proof-search is a fundamental enabling technology throughout the computing sciences. Moreover, the reductive view of inference represents an alternative view of logic, just as fundamental as the deductive one, which is largely undeveloped.

So far, the theory of proof-search has developed mostly along prooftheoretic lines but using many type-theoretic techniques. The utility of type-theoretic methods suggests that semantic methods of the kind found to be valuable in the semantics of programming languages should be useful in tackling the main outstanding difficulty in the theory of proof-search,

i.e., the representation of intermediate stages in the search for a proof. The space of searches is much larger than the space of proofs: An adequate semantics would represent both the space of searches and the space of proofs and give an account of the recovery of proofs (which are extensional objects) from searches (which are more intensional objects). It would distinguish between different proof-search strategies and permit analyses of their relative merits.

This seminar helps to establish a program to build such a semantics. To this end, we propose the following foci for the seminar:

- Reductive vs. deductive logic: their logical, mathematical and computational properties;
- Proof-search in type-theoretic languages: the role of typing constraints during proof-search;
- Proof- and model-theoretic analyses of search spaces: the search-oriented counterparts to traditional proof theory and model theory;
- Intensional semantics for proof-search: specific intensional and computational models based on structures such as games, continuations and realizability;
- Applications of proof-theoretic and semantic techniques to the design and implementation of theorem provers.

The Seminar

The seminar was lively and friendly, with many people commenting that they found the exposure to some new ideas quite stimulating. It was particularly pleasing that there was little or no tendency among the participants to form into subgroups: Everyone talked to everyone else.

Several broad themes may be identified in the given lectures:

- Foundational issues: Basic questions about the meaning and mathematical semantics of search spaces and search-objects;
- Logic programming: Issues in semantics and pragmatics;
- Type theory and interactive theorem proving: Issues in the formulation and representation of problems;

- Tableaux and counter-models;
- Syntactic methods: Optimizing the execution dynamics of search engines via the logical properties of the target system;
- Applications: To formal mathematics, to logic programming and to verification of Java bytecode!

Abstracts selected by the Dagstuhl News editor:

Modelling backtracking

Eike Ritter, University of Birmingham

The work presented in this talk is part of a bigger project, which intends to give semantics to proof search. In this talk we present some general steps which are necessary to achieve such a semantics, and focus on one aspect, namely how to model backtracking in intuitionistic logic via continuations.

The first step consists of giving semantics to partial, possibly incompleteable proofs. We use polynomial categories for this purpose. The universal property of these categories ensures that a partial proof can be completed to a proof if and only if one can find a substitution consisting only of ground terms for the indeterminates in the polynomial category . The left-rules of the sequent calculus force us to consider also a Kripke-style semantics where the information contained in the Kripke-worlds is the substitution arising from modelling the implication left-rule.

In this paper we model backtracking by embedding intuitionistic logic into classical logic. Hence in a second step we extend these polynomial categories and the Kripke-semantics to the $\lambda\mu\nu$-calculus, a term calculus for classical logic. In this semantics, a switch in the focus on the right-hand side corresponds to applying a continuation in functional languages.

When we embed LJ-proofs with backtracking into LK and translate these proofs into the semantics we have developed, we realize that backtracking in LJ-proofs gives rise to switching the right-hand side in the LK-proofs resulting from the embedding, and hence to continuations.

This is joint work with David Pym.

Ontological status of paraproofs

Thomas Streicher, University of Darmstadt

Both in proof-search and proof theory certain paraproofs show up, i.e., derivations where some leaves are not justified by axioms but by authority. These justifications by authority rather serve the purpose of error elements known from programming. These paraproofs do not serve in general the purpose of establishing truths but rather are used for testing the real proofs.

This is basic to Girard's Ludics programme. But paraproofs also appear in recent work of Krivine and Danos on realizability for classical AF_2 and in Curien and Herbelin's Duality of Computation where paraproofs appear as continuation terms. We leave it as a question for future investigations to find out whether paraproofs may provide a bridge between the fields of proof search and continuation semantics.

9 Product Family Development

Seminar No. **01161** Report No. **304** Date **16.04.–20.04.2001**
Organizers: Günter Böckle, Paul Clements, Henk Obbink, Klaus Pohl, Dieter Rombach

Software Engineering is an expensive and time consuming task. One strategy for reducing the effort in application building is reuse of work products from other projects. The product family approach promises to maximize reuse in a systematic way. Product family development focuses on the creation and maintenance of a whole set (family) of software products. It has recently gained much interest in various application domains including electronic commerce, information systems, medical systems and telecommunication systems. Product family development differentiates between the creation and the maintenance of system assets (development artefacts) which are common to the various application systems and the assets that are specific to particular applications. In contrast, research in traditional software engineering disciplines has mainly focused on "one of a kind" systems.

The principal ideas and solutions developed for "one of a kind" systems in industry and research may still be applicable to product family development. But, product family development requires an adjustment

and extension of those concepts, especially in the areas of requirements engineering, software architecture and components. Software architectures and distributed components, for example, have to be built under the premise that evolution of the product family is inevitable and they thus have to provide solutions for actually mapping variable parts onto interfaces and code. Or, requirements engineering must take the results of the domain analysis into account when defining the application specific requirements. User needs should, whenever possible, be mapped to requirements already satisfied by the core architecture to guarantee a successful reuse of other product family assets.

This Dagstuhl Seminar convened twenty-six leading practitioners and researchers from various disciplines to cross-examine the effectiveness and the efficiency of product family based software system development. The seminar was mainly organised by the EUREKA/ ITEA Project ESAPS (Engineering Software Architectures, Processes and Platforms for System Families) in cooperation with the SEI (Software Engineering Institute, Carnegie Mellon University, PA, USA).

After overview talks on "the American view" on software product lines (by Linda Northrop, SEI, USA) and "the European view" on software product lines (by Frank v. d. Linden, Philips, Netherlands), requirements engineering (by Klaus Pohl, University of Essen, Germany), architectures for product families (by Paul Clements, Carnegie Mellon University, USA) , variability in product families (by David M. Weiss, Avaya Communication, USA) , scoping of product families (by Peter Knauber, Frauenhofer IESE, Germany), three main topics for the seminar where identified as a result of a brainstorming session and discussed in parallel working groups:

1. Product line adoption strategies and convincing business cases;
2. Managing variability in space and time for software intensive systems;
3. Economics and marketing issues of product lines

10 Computational Cartography and Spatial Modelling

Seminar No. **01191** Report No. **305** Date **06.05.–11.05.2001**
Organizers: M. Worboys, R. Weibel, M. van Kreveld

The Dagstuhl seminar on computational cartography and spatial modelling is the third in a series of seminars where computer scientists and spatial scientists are brought together. It is the first time that the topic of spatial modelling was added to extend the scope of the seminar. The exchange of ideas and research between computer science and spatial science is essential to advance research in this interdisciplinary area.

The seminar was attended by twenty-six participants from various countries. Most participants were affiliated at universities, but there were also participants from companies and institutions, among which national mapping agencies. The main topics addressed during the seminar were:

- Cartographic methods: automated map labeling and automated map generalization remain crucial issues both in research and software for automated cartography.
- Spatial and spatio-temporal modelling: the definition of formal, conceptual models for time and space which underlie spatial data and its use in geographic information systems.
- Terrain modelling: defining and computing realistic terrain models, and computations on existing digital elevation models.
- Techniques from computational geometry: techniques and topics new in computational geometry that can be of use in geographic information systems.

The surroundings and the atmosphere provided the ideal conditions for a lot of interaction and discussions among the participants. The organisers hope that the seminar will be followed by a fourth on the topic.

Abstracts selected by the Dagstuhl News editor:

Pedestrian Guidance Information from City Maps

Stephan Winter

Location-based services are a hot topic in research and development since telecom providers are looking for revenues for their investments in 3G-technology. Ubiquitous access combined with mobility of people creates a new market.

Pedestrian guidance services are a specific kind of location-based services. They are based on positioning techniques for tracking the user, and access to a variety of distributed information / service providers, some of them contributing spatial information, some of them contributing content information.

The talk deals with the creation of a graph representing the network of places open for pedestrian navigation in urban environment. Considering different sources and different types of sources it is discussed in detail how a graph can be constructed from two-dimensional spatial objects contained in multi-purpose city maps. Spaces are classified for their accessibility first, then for their shape. Two different kinds of abstracted linearization are presented and compared: center linearization and exit linearization. Both resulting graphs show useful properties for instructing pedestrians; and both are sufficiently abstract from geometric details an autonomous pedestrian is not interested in.

In the outlook the relevance of the graph properties for the construction of route instructions is discussed, and the problem of tracking along a relatively generalized route is mentioned. Topics like user interface design, user profiling, or individual adaptation of the guidance service are not considered so far.

11 Algorithms and Number Theory

Seminar No. **01201** Report No. **306** Date **13.05.–18.05.2001**
Organizers: J. Buhler, H. Niederreiter, M.E. Pohst

This seminar on number-theoretical algorithms and their applications was the fourth on this topic at Dagstuhl over the last 10 years. This year 45 people from 14 countries participated.

One of the major goals of these has been to broaden interactions between number theory and other areas. For instance, there has been an ef-

fort to bring together people developing the theory of efficient algorithms with people actually writing software. There has also been continuing interest in cryptography, and this year almost a third of the talks were on algebraic curves, most with an eye to applications in cryptography. The use of elliptic curves in cryptography seems to be well understood by now, and the focus is on speeding up the algorithms, whereas the research on the use of hyperelliptic curves is more focused on developing the mathematical foundations of the field.

Many other talks focused on more classical topics of algebraic number theory, such as finding divisor class groups of function fields, finding Galois groups, and investigating class groups and their heuristics.

The remaining talks covered a wide variety of problems in algorithmic number theory, including hardware implementations of arithmetic over fields of characteristic 2, a parallel sorting algorithm with applications to integer factorization, finding solutions to diophantine equations, and factoring polynomials in various domains.

The variety of topics was stimulating to the audience (though it did make the organizers' task of grouping the talks more difficult!). The reaction of the participants was quite positive and we believe that we succeeded in having an effective meeting that was able to appeal to a broad audience. We made sure to allow for adequate breaks between sessions, and there were many opportunities for discussions that the participants took advantage of. The pleasant atmosphere of Schloss Dagstuhl once again contributed to a very productive meeting.

12 Software Visualization

Seminar No. **01211** Report No. **307** Date **20.05.–25.05.2001**
Organizers: Stephan Diehl, Peter Eades, John Stasko

It is often said that humans have never before created any artifacts which are as complex as today's software systems. As a result creating, maintaining, understanding and teaching software is a challenging task. Software is neither matter nor energy, it is just a kind of information. Sometimes the representation and the information itself are confused. Software visualization is concerned with visually representing different aspects of software including its structure, execution and evolution.

So far, research on software visualization was mostly motivated by its potential to support teaching. Many systems have been developed to facilitate the production of algorithm animations.

At Dagstuhl software engineers and re-engineers repeatedly argued that there is a strong need for software visualization in their areas. Here further research includes the use of techniques from information visualization to display software metrics, graph layout and graph animations to show the structure and changes in software systems and program animation for debugging.

At the seminar more than 50 researchers from all around the world discussed the state-of-the-art as well as challenging questions for the future of software visualization. The program included 38 presentations and 15 system demonstrations, as well as several sessions for group discussions.

Participants of the seminar volunteered

- to compile a post seminar proceedings, which is to be published as a Springer LNCS state-of-the-art survey.
- to create a repository with algorithm animations and software visualization tools
- to initiate an international conference series on software visualization.

We feel that the seminar was a seminal event. The future will tell whether it reached its ambitious goals to form a community and raise awareness of software visualization as a challenging and important research field of its own.

13 Can Formal Methods Cope with Software-Intensive Systems?

Seminar No. **01221** Report No. **308** Date **27.05.–01.06.2001**
Organizers: Stefan Jänichen, Jeff Kramer, Michel Lemoine, Martin Wirsing

During the last years practical Software Engineering techniques are used more and more to conduct systematic and rigorous development of large software systems. UML has become the standard notation for guiding and documenting Software Engineering projects. CASE tools of today offer not only the UML notation but also are able to generate code templates and to support round trip engineering between class diagrams and program code. However, used in practice they do not support well the early phases of software development; they still lack analysis and validation methods for requirements and design specifications which are easily connected to the implementation phase.

Formal techniques have undergone a steep development during the last years. Based on formal foundations and deep theoretical results, methods and tools have been developed to support specifications and design of software systems. Model-based and algebraic specifications, abstract state machines, CSP and CCS, temporal logics, rewriting techniques, finite automata, model checking and many other formalisms and verification techniques have been applied to non-trivial examples and are used in practice e.g. for the development of safety critical systems. Several case studies have been proven to be useful for validating and evaluating formal software development techniques. Case studies tackle the development in the small such as the production cell, the steam boiler and the memory cell. What is missing is a comparison of the development in the large. How do known formal techniques scale up? How do they cope with aspects such as architecture, component ware, distribution, mobility reconfiguration?

The aim of this workshop was to contribute to the field of Experimental System Engineering by proposing a case study for system development which allows one to compare different formal techniques in their abilities to specify, design, analyze and validate large software-intensive systems. The case study addresses the actual problem of controlling autonomous trains and systems and contains features such as local control in a distributed system, synchronous and asynchronous communication, heterogeneous components, and optimization problems.

During the workshop the solutions for the case study were presented and discussed by the participants. Also related work on formal and semi-formal approaches to system development was presented.

Abstracts selected by the Dagstuhl News editor:

Methods and Experiments

Stefan Jähnichen, Technical University, Berlin

The talk gives some ideas on what a method should be and more important, what a method should contain. Although there are good reasons to apply systematic development techniques in a systematic manner, two examples are given for which even in an academic environment, the processes were not applied systematically and consistently. The reasons for this observation are found in the complexity of the developments and in the difficulty to identify the requirements correctly. The two examples are a complex scheduling problem for a large railway network and the control system for a new satellite. It is expected that the further development of the examples is accompanied by a formal treatment to improve reliability, security, and fault tolerance.

14 Design and Analysis of Randomized and Approximation Algorithms

Seminar No. **01231** Report No. **309** Date **03.06.–08.06.2001**
Organizers: Martin Dyer, Mark Jerrum, Marek Karpinski

Overview

The Workshop was concerned with the newest development in the design and analysis of randomized and approximation algorithms. The main focus of the workshop was on two specific topics: *approximation algorithms for optimization problems*, and *approximation algorithms for measurement problems*, and the various interactions between them. Here, new important paradigms have been discovered recently connecting probabilistic proof verification theory to the theory of approximate computation. Also, some new broadly applicable techniques have emerged recently for designing efficient approximation algorithms for a number of computationally hard optimization and measurement problems. This workshop has addressed the above topics and also fundamental insights into the new paradigms and design techniques. The workshop was organized jointly with the RAND-APX meeting on approximation algorithms and intractability and was partially supported by the IST grant 14036 (RAND-APX).

Motivation

Most computational tasks that arise in realistic scenarios are intractable, at least if one insists on exact solutions delivered with certainty within a strict deadline. Nevertheless, practical necessity dictates that acceptable solutions of some kind must be found in a reasonable time. Two important means for surmounting the intractability barrier are *randomized computation*, where the answer is optimal with high probability but not with certainty, or *approximate computation*, where the answer is guaranteed to be within, say, small percentage of optimality. More often than not, these two notions go hand-in-hand.

The seminar will be concerned with these phenomena. It will address the newest development in the design and analysis of randomized approximation algorithms, and the new fundamental insights into computational approximate feasibility, optimality, and the intractability of various computational problems. The main focus of the workshop is to be on two specific topics and the various interactions between them. The specific topics are the following:

- **Approximation algorithms for optimization problems.**

  Randomization and de-randomizing techniques play a major role here, both in *positive* (upper bounds) and *negative* (lower bounds) results. It features for example in the "rounding" step of approximation algorithms based on linear or semidefinite programming relaxations; it is also at the heart of the theory of *probabilistically checkable proofs* (PCPs) that is the basis for the recent non-approximability results. A number of very significant new results were obtained here recently.

- **Approximation algorithms for measurement problems.**

  The word "measurement" here is used to distinguish a class of problems – determining the cardinality of combinatorially or computationally defined sets, volume, expectation of random variables on configurations of complex systems, etc. – which are very different in flavor of the optimization problems. This theme is less developed than the previous one, but significant progress is currently being made, both in design of efficient approximation algorithms, and in proving the first approximation lower bounds based on the

PCP-techniques mentioned before. It is aimed here at investigating further fundamental and intrinsic connections between the efficiency of approximating optimization problems and the efficiency of approximating measurement problems.

The main goal of the seminar was to bring together researchers working in the area of approximation algorithms and approximation complexity of computational problems, and focus on the newest developments (including practical implementations) within, and also in between the above main themes.

Abstracts selected by the Dagstuhl News editor:

Optimal myopic algorithms for random 3-SAT

Gregory Sorkin, Mathematical Sciences Dept., IBM T.J. Watson Research Center

3-SAT is a canonical NP-complete problem: satisfiable and unsatisifiable instances cannot generally be distinguished in polynomial time. However, random 3-SAT formulas show a phase transition: sparse instances are almost always satisfiable, and dense ones almost always unsatisfiable.

Proofs of the satisfiability of sparse instances have come from analyzing simple heuristics: the better the heuristic analyzed, the denser the instances that can be proved satisfiable with high probability. To date, the useful heuristics have all been simple extensions of unit-clause propagation, all expressible within a common framework, and analyzable in a uniform manner by employing differential equations.

Here, we determine optimal algorithms expressible in that framework, establishing an improved density bound. We extend the analysis via differential equations, and make extensive use of a new optimization problem we call "max-density multiple-choice knapsack". The structure of optimal knapsack solutions elegantly characterizes the choices made by an optimal algorithm. Joint work with Dimitris Achlioptas.

15 Management of Metacomputers

Seminar No. **01241** Report No. **310** Date **10.06.–15.06.2001**
Organizers: Francine D. Berman, Alexander Reinefeld, Uwe Schwiegels-

hohn

Background and Motivation

The success of the Internet along with the worldwide growing number of high-performance computers has led to the concepts of *metacomputing* or, more recently, *computational grids*. In principle, a metacomputer can be considered as an extension of a distributed computer with a variety of geographically dispersed resources, such as supercomputers, storage systems, data sources and special devices. Ideally, such a metacomputer is seen as a single unified resource by the user. It typically consists of various architectures with different application software. Those architectures usually belong to different owners and are accessed from a large number of independent users.

As already mentioned the distributed nature of an ideal metacomputer environment is transparent to the user, that is, he or she only needs to describe the constraints connected with a job while the system selects the most suitable machine for the execution of this job. This selection process may be subject to a large variety of different constraints including access restrictions, user priorities, machine workload, job characteristics and user preferences. In addition, the metacomputer structure may change due to maintenance shut downs or temporary failures of sub-systems. It is the task of the management software to handle those problems, that is, to provide the desired transparent access to the users while at the same time considering any special requests from users and owners. Therefore, the management software is a key component of a metacomputer.

Contents of the Seminar

The architecture and the methods of such a management software were the focus of this Dagstuhl seminar. The participants explored and analyzed the design, implementation, and deployment of metacomputer management systems. Eighteen talks were given on the various aspects and the state-of-the-art of metacomputer management.

The first day of the seminar (Monday) was devoted to the topic of scheduling. Various projects, concepts, and new approaches have been presented by Jon Weissman, Dick Epema, Uwe Schwiegelshohn, Larry

Carter, and Volker Sander. The day was concluded with a discussion on current aspects and future developments in the field.

The talks of Marian Bubak (given by Roland Wismüller), Barton Miller, and Arnaud Legrand on Tuesday focused on performance issues, program development and security issues in grid environments. In the afternoon, Lennart Johnsson discussed aspects of grid application tools, and thereafter several large scale projects were briefly presented by Volker Sander (Globus), Thilo Kielmann (DAS), Florian Schintke (Datagrid), André Merzky (Cactus), and Steve Chapin (Legion). The pros and cons of these projects were compared and vividly discussed until late in the evening.

On the third day, various topics of grid infrastructure, grid components, and application specific grids were presented by André Merzky, Thilo Kielmann, and Domenico Talia. In the afternoon, participants took a hike through the nearby forests, discussing various research topics in a leisurely surrounding at an excellent weather.

Thursday was filled with talks on grid infrastructure and tools. Steve Chapin, Ramin Yahyapour, Jean-Marc Nicod, Alexander Reinefeld, Roland Wismüller, Florian Schintke, and Volker Lindenstruth presented their work on the various components used to build metacomputer environments. Again, some of these topics were discussed in more detail until late in the evening.

16 Stochastic Methods in Rendering

Seminar No. **01242** Report No. **311** Date **10.06.–15.06.2001**
Organizers: Mateu Sbert, Werner Purgathofer, Pete Shirley

Stochastic methods have become indispensable tools in computer graphics, and more specifically in rendering, since the mid 1980s. Now these techniques are used in all subfields of rendering, and are part of every major commercial rendering package.

As with most computer graphics research, work from related fields such as radiative heat transfer and neutron transport have been modified and applied in rendering. However, once the basic technique is introduced to computer graphics, its researchers often improve on it in such a way that sometimes it is transferred back to the field from where

it was borrowed, as it has happened with some radiative heat transfer methods. A second closely related topic is quasi-Monte Carlo integration and randomized quasi-Monte Carlo integration.

Although specific conferences exist separately on Monte Carlo techniques and rendering, they are both too general to give an opportunity to Monte Carlo researchers in rendering for meeting and evaluating the specific impact of these techniques on their field. The purpose of this first seminar focused on "Stochastic Methods in Rendering" was thus to give the opportunity to evaluate the past, present and future perspectives of the use of stochastic and quasi-Monte Carlo techniques in rendering.

The workshop had to be scheduled for the same week as another seminar at Schloss Dagstuhl, and therefore the number of participants was restricted to be low (32 people from 10 countries). However this reduced number turned out ideal to stimulate real discussions. Many well established specialists came together with several young researchers, mostly PhD students. The overall highlight of the seminar was the presence of Professor John Halton, a pioneer of the quasi-Monte Carlo method. It was very interesting to follow the discussions between people, once they had been made aware of tricks and hints used by others in completely different contexts. Most participants profited a lot from this exchange of normally not mentioned details in Monte Carlo and quasi-Monte Carlo algorithms.

"Stochastic Methods in Rendering" are concerned with numerically computing the integrals underlying the generation of synthetic images from digital data. These integrals very often are high-dimensional and usually the integrands are discontinuous, too. So only methods based on stochastic tools provide appropriate algorithms for the simulation of light transport. 21 talks of high quality were given on that subject. Many talks described the application of stochastic methods in diverse algorithms, but some also described algorithms that were developed for stochastic use. A few others were rather on principles of Monte Carlo and quasi-Monte Carlo methods. Among others the following topics were discussed during and after the talks:

- theoretical comparisons of radiosity algorithms,
- utilization of random walk methods for global illumination,
- importance sampling and optimal ray distribution techniques,

- rendering animations,
- stochastic methods for real-time rendering, and
- complex natural objects.

The permanent discussion, whether random or quasi-random methods are better, resulted in a tutorial on correlated sampling by Alexander Keller, which stimulated a very interesting discussion. The main facts were that the lower bound of Monte Carlo integration is obtained using correlated sampling with the separation of the main part and the multi-level method of dependent tests, that even multiple importance sampling can arbitrarily fail due to the problem of insufficient techniques, and that correlated stratification is intrinsic to (randomized) quasi-Monte Carlo techniques. A then obvious but very important conclusion was that correlated sampling in rendering is worth further profound investigations.

17 Graph Decompositions and Algorithmic Applications

Seminar No. **01251** Report No. **312** Date **17.06.–22.06.2001**
Organizers: A. Brandstädt, J.P. Spinrad

There are many notions of graph decomposition which arise in the literature. Some decompositions involve decomposing a graph using separators of special types (balanced or polynomially bounded, star cutsets, clique cutsets), others involve identification of special sets (substitution or splits), while others involve tree decomposition (treewidth, cliquewidth, branchwidth) or tree composition (Cartesian product, lexicographic product).

These decompositions are of fundamental importance for solving optimization and recognition problems on classes of graphs. For example, substitution decomposition is closely related to such problems as solving problems expressible in monadic second order logic quantifying over vertices and/or edges and comparability graph recognition and optimization. Treewidth and its generalizations are of special importance due to the Robertson-Seymour results on tree decomposition and existential proof

of existence of algorithms. Clique cutsets and star cutsets are fundamental tools used in the study of chordal and perfect graphs. Particular tools for working with these decompositions, such as partition refinement and lexicographic breadth first search, have recently been improved and generalized in this context.

The second Dagstuhl seminar on Graph Decompositions and Algorithmic Applications was designed to bring together researchers working on a variety of aspects of graph decomposition. Talks were given reporting on recent results concerning the cliquewidth of graphs and its algorithmic use, the connection between cliquewidth and treewidth, studying special classes of graphs, new decomposition techniques and optimization algorithms, and data structures which allow faster decomposition algorithms.

Abstracts selected by the Dagstuhl News editor:

On the relationship between clique-width and treewidth

Derek Corneil, Udi Rotics

Treewidth is generally regarded as one of the most useful parametrizations of a graph's construction. Clique-width is a similar parametrization that shares one of the powerful properties of treewidth, namely: If a graph is of bounded treewidth (or clique-width), then there is a polynomial time algorithm for any graph problem expressible in Monadic Second Order Logic, using quantification on vertex sets (in the case of clique-width you must assume a clique-width parse expression is given). In studying the relationship between treewidth and clique-width, Courcelle and Olariu showed that any graph of bounded treewidth is also of bounded clique-width; in particular for any graph G with treewidth k, the clique-width of $G \leq 4 * 2^{k-1} + 1$. (Johansson's result on NLC width shows that the "+" is not needed.)

In this paper we improve this result to the clique-width of $G \leq 3 * 2^{k-1}$ and more importantly show that there is an exponential lower bound on that relationship. In particular, for any k, there is a graph G with treewidth $= k$ where the clique-width of $G \geq 2^{\lfloor k/2 \rfloor - 1}$

Finding houses and holes in graphs

Chính T. Hoàng, R. Sritharan

A *house* is the complement of an induced path on five vertices. A *hole* is an induced cycle on five or more vertices. A *domino* is the cycle on six vertices with a long chord. A graph is HH-free if it does not contain a house or a hole. A graph is HHD-free if it does not contain a house, or a hole, or a domino.

We present $O(n^3)$ algorithms to recognize HH-free graphs and HHD-free graphs. The previous best algorithms for the problems run in $O(n^4)$ time.

18 Information and Simulation Systems for the Analysis of Gene Regulation and Metabolic Pathways

Seminar No. **01261** Report No. **313** Date **24.06.–29.06.2001**
Organizers: Ralf Hofestädt, Nikolay Kolchanov, John Reinitz

The third Dagstuhl Seminar for Information and Simulation Systems for the Analysis of Gene Regulation and Metabolic Pathways was held from June, 24 to 29, 2001. It was a multidisciplinary seminar with participants from 11 different countries. Schloss Dagstuhl workshops in general emphasize computer science, and we are delighted to focus on the rapidly developing links between biosciences and computer sciences. The 2001 meeting is a sequel to the 1995 and 1998 meeting on the similar topic.

Molecular biology and biotechnology have begun to focus sharply on the problem of gene regulation. This problem is inescapable, because no open reading frame (ORF) will be expressed without the appropriate regulatory sequences. Moreover, some genes code for proteins whose function is to turn other genes on and off. Groups of these genes form networks with complex behaviors. These networks control other genes whose protein products catalyze specific biochemical reactions, and the small molecules which are substrates or products of these reactions can in turn activate or deactivate proteins which control transcription or translation. For that reason, gene regulation can be said to indirectly control biochemical reactions in cellular metabolism, and cellular metabolism

itself exerts control on gene expression. For these reasons, the interdependent biochemical processes of metabolism and gene expression can and should be interpreted and analyzed in terms of complex dynamical networks. Hence modeling and simulation are necessary. Two earlier Dagstuhl seminars (1995 and 1998) have already dealt with modeling and simulation of biochemical networks. Both sought to bridge two divides by both bringing together scientists in the disciplines of gene regulation and metabolic pathways, and within and across both of these areas bringing together experimentalists and theoreticians. Often there had been little previous contact among these groups, but clearly the integration of metabolic and gene expression models as well as the cooperation of theorists and experimentalists is essential in order to solve these complex problems.

Apart from theoreticians and experimentalists, a third group has emerged since 1995 which is centered around databases and the internet. Many molecular biologists turned towards informatics and systematically collected results relating to specific problems. These data have been and will be stored systematically in specific databases, which nowadays are accessible via the Internet. Recently many firms have been founded which provide data essential for the solution of scientific and industrial problems, and even more importantly the corresponding infrastructure. As a result, there are databases available via the Internet for all known sequenced genes (e.g. EMBL), proteins (e.g. SWISS-PROT, PIR, BRENDA), transcription factors (TRANSFAC), biochemical reactions (KEGG) and signal induction reactions (TRANSPATH, GeneNet). Beyond databases, simulators for metabolic networks which employ most of the currently popular modeling methods are also available via the Internet. In addition to the classical methods of differential equations, discrete methods have become quite important. Examples are the object-oriented approach, rule-based systems, Petri Nets, graphs, and Boolean nets.

These recently implemented tools on the Internet are the basic components of the informatic and analytical infrastructure of biotechnology. Clearly the next evolutionary stage of development will be the implementation of integrated molecular information systems (e.g. SRS). The first step to reach that goal is the integration of databases under a specific biological perspective. The next step will be user-defined molecular information fusion. Up to now, there are no standard tools available in order to successfully separate both methods and databases. Exactly for

that reason it is imperative to develop uniform intersections at this stage. To discuss properties of these intersections was one major issue of this seminar.

19 Link Analysis and Visualization

Seminar No. **01271** Report No. **314** Date **01.07.–06.07.2001**
Organizers: Ulrik Brandes, David Krackhardt, Roberto Tamassia, Dorothea Wagner

The purpose of this seminar was to introduce to each other researchers working on different aspects and applications of link analysis and visualization in order to strengthen the algorithmic foundations of this rapidly emerging, highly interdisciplinary, field.

Link analysis explores associations among entities of arbitrary type. It is increasingly recognized as a fruitful extension of categorical approaches to data analysis in a fast growing number of application domains. Example applications are the analysis of linkages on the Web (search engines, site maps), network traffic monitoring (Web caching, public transport), data mining (e-commerce, telecommunications services), social network analysis (social structures, policy making), text analysis (coreference, cocitation), decision support (financial markets, logistics), or fraud detection (money laundring, calling cards).

Typical objectives in these applications are the identification of central or bottleneck entities, structural patterns and trends, effective modifications, hidden or missing data, substructures, appropriate levels of aggregation, similarities among data sets, etc., and visualization has proven crucial in assisting humans to comprehend complex relational structures and identify unexpected patterns.

Abstracts selected by the Dagstuhl News editor:

David Krackhardt's Interpretation of Valdis Krebs' Consulting Using Network Graphs

David Krackhardt (Carnegie Mellon University)

Valdis Krebs is one of the most successful and prominent organizational consultants applying network analysis to business organizations today.

This presentation of his work centered on three examples from his experience in the field. First, his graphs show how managers prefer to think in terms of formal organizational units (departments) and to draw pictures of the informal organization using circles representing these units. With a spring embedder, however, these pictures are much clearer and more informative, even to these same managers. With Valdis' help, IBM studied a pharmaceutical firm, using four relational questions. They discovered that the firm had many holes in its network and concluded that efforts should be made to plug those holes. Also, they could see that one department was completely self-absorbed and that they needed to make bridges to the rest of the organization to generate new ideas.

The second example followed a merger by two firms. Network pictures were used to monitor the progress of integration of this merger. It was clear from the pictures that top management was not integrating very well – most of management was associating only with people in their original company.

Third, he provided an example of an inter-firm analysis, showing how the high tech industry is forming strategic alliances with many different kinds of firms in an attempt to expand and ensure profitability in their businesses.

Finally, Krackhardt provided an example of how the dynamics of a firm's response to a unionization attempt could be understood by looking at the social network ties in the firm. One individual was chosen to lead the group, but he was ineffective because he was completely isolated from the group in the friendship network. Second, a key player (Chris), who could have been very influential in getting the people to support the union, was over-embedded in a mass of strong ties that prevented him from speaking his mind freely on the union issue. Finally, Krackhardt showed how the complex network picture could be reduced to a simple but powerfully explanatory picture of the structure by performing a structural equivalence role analysis on Simmelian (co-clique) ties in the firm.

20 Inference Principles and Model Selection

Seminar No. **01301** Report No. **315** Date **23.07.–27.07.2001**
Organizers: Joachim Buhmann, Bernhard Schölkopf

The core problem of statistics and machine learning addresses the question how can we efficiently find a statistical model to describe empirical data. Classical statistical approaches to solve this problem have been complemented during the last 15 years by Neural Computation, a very promising strategy to data analysis. The Dagstuhl seminar on "*Inference Principles and Model Selection*" — the fourth in a series of Machine Learning and Neural Computation workshops in 1994, 1997, 1999 and 2001 — was intended to review this exciting development of the field and to discuss the foundation of statistical and computational learning theory with its deep (and still unresolved) questions. The participants represented all of the involved disciplines from statistics and computer science to information theory and philosophy. The burning question of many participants if there exist notions of inference studied in philosophy that machine learning has overlooked so far came up in several sessions and especially in the first tutorial on Philosophical Foundations (Matthias Hild). Three pioneers of the field, Sun-ichi Amari, Phil Dawid and Vladimir Vapnik provided valuable insights how the field of learning machines developed from the sixties up to today and what kind of challenges are lying still ahead of us.

What have been the main conclusions of the seminar?
In contrast to the previous seminars, this workshop with its tutorials and short position statements (rather than conference style talks) forced the participants to concentrate on conceptual issues with as little obstruction as possible by technical details. Common ground between Bayesian inference, statistical and computational learning theory and logical approaches to inference as well as concepts from information theory have been observed and widely discussed.

The final discussion session summarized the following open issues of the field:

1. Tali Tishby reminded us that learning and information extraction goes beyond the issues of sample fluctuations which are extensively studied in the Computational Learning Theory community. What are the correct inference principles to detect structures hidden in data?

2. How can we evaluate learning principles and algorithms? How should we design good experiments?

3. Is model selection or model combination more effective in structure detection?

4. How can we find more characterization results for learning algorithms? What is an appropriate size of the validation set?

5. How should we proceed in non i.i.d. situations where data are dependent? How can the concepts from classification and regression be extended to time series analysis and to Markov random fields?

6. What is the correct number of inference levels?

Most of these questions will stay with us for the next decades but this workshop has raised the awareness of all participants which parts of machine learning and neural computation are based on fundamental principles and where we still have to discover such a solid foundation.

Remark: The abstracts and links to slides are available at `www-dbv.cs.uni-bonn.de/dagstuhl01`.

21 Parameterized Complexity

Seminar No. **01311** Report No. **316** Date **29.07.2001 - 03.08.2001**
Organizers: Rod G. Downey, Michael R. Fellows, Rolf Niedermeier, Peter Rossmanith

Parameterized complexity is a new and promising approach to the central issue of how to cope with problems that are NP-hard or worse — as is so frequently the case in the natural world of computing. The key idea is to isolate some aspect(s) or part(s) of the input as the *parameter*, and to confine the seemingly inevitable combinatorial explosion of computational difficulty to an additive function of the parameter, with other costs being polynomial (called FPT complexity). An example is the NP-complete VERTEX COVER ("conflict resolution") problem that is now known to be solvable in less than $1.29^k + kn$ steps for conflict graphs of size n. This algorithm works well for $k \leq 200$ and has several applications in computational biology.

Many important "heuristic" algorithms currently in use are FPT algorithms, previously unrecognized as such. Type-checking in ML provides another example. Although complete for EXPTIME in general, it is

solved in practice in time $2^k + n$ for programs of size n, where the k is the nesting depth of declarations. Although many naturally parameterized problems are in FPT, some are not. The rich positive toolkit of novel techniques for designing and improving FPT algorithms is accompanied in the theory by a corresponding negative toolkit that supports a rich structure theory of parametric intractability. But the real excitement is in the rapidly developing systematic connections between FPT and useful heuristic algorithms — a new and exciting bridge between the theory of computing and computing in practice.

The organizers of the seminar strongly believe that knowledge of parameterized complexity techniques and results belongs into the toolkit of every algorithm designer. The purpose of the seminar was to bring together leading experts from all over the world, and from the diverse areas of computer science that have been attracted to this new framework. The seminar was intended as the first larger international meeting with a specific focus on parameterized complexity, and it hopefully serves as a driving force in the development of the field.

Abstracts selected by the Dagstuhl News editor:

Probabilistic 3-SAT Algorithms

Uwe Schöning

We present a series of 3 algorithms for 3-SAT (which can be generalized to k-SAT) based on the concept of local search from some randomly selected initial assignment, and restart if no satisfying assignment is found. The first version uses random initial assignments and a deterministic backtracking procedure to search for a satisfying assignment within Hamming distance $n/4$ from the initial assignment. It achieves the bound (1.5^n) (where n is the number of variables). The second algorithm replaces the backtracking search by a random walk, and using a Markov chain analysis (gambler's ruin problem) one can show the improved bound $((4/3)^n)$. The third algorithm, finally, looks out for "independent" clauses and chooses the initial assignment for variables in independent clauses in a biased way. It can be shown that the obtained bound is (1.3301^n).

22 Dependent Type Theory Meets Practical Programming

Seminar No. **01341** Report No. **317** Date **19.08.–24.08.2001**
Organizers: Gilles Barthe, Peter Dybjer, Peter Thiemann

Modern programming languages rely on advanced type systems that detect errors at compile-time. While the benefits of type systems have long been recognized, there are some areas where the standard systems in programming languages are not expressive enough. Language designers usually trade expressiveness for decidability of the type system. Some interesting programs will always be rejected (despite their semantical soundness) or be assigned uninformative types.

There are several remedies to this situation. We argue that dependent type systems, which allow the formation of types that explicitly depend on other types or values, are one of the most promising approaches. These systems are well-investigated from a theoretical point of view by logicians and type theorists. For example, dependent types are used in proof assistants to implement various logics and there are sophisticated proof editors for developing programs in a dependently typed language.

To the present day, the impact of these developments on practical programming has been small, partially because of the level of sophistication of these systems and of their type checkers. Only recently, there have been efforts to integrate dependent systems into intermediate languages in compilers, for example, the TAL compiler (Morrisett and others), and actual programming languages, for example, Cayenne (Augustsson) and DML (Xi and Pfenning). Additional uses have been identified in high-profile applications such as mobile code security. For example, proof carrying code (Necula and Lee) relies on a dependently typed lambda calculus to encode proof terms.

Now the time is ripe to bring together researchers from the two communities (type theorists and programming experts), and to further cross-fertilization of ideas, techniques and formalisms developed independently in these communities. In particular, the seminar shall make researchers in programming languages aware of new developments and research directions on the theory side; point out to theorists practical uses of advanced type systems and urge them to address theoretical problems arising in emerging applications.

The need for such a seminar became clear during the first international Workshop on Dependent Types in Programming, held in Gteborg in March 1999. A second international Workshop on Dependent Types in Programming 6 has been held in Ponte de Lima in July 2000, but it is hard to discuss the problems pointed out above in a one-day workshop.

23 Foundations of Semistructured Data

Seminar No. **01361** Report No. **318** Date **02.09.–07.09.2001**
Organizers: Alberto Mendelzon, Thomas Schwentick, Dan Suciu

Traditional database systems rely on an old model: the relational data model. When it was proposed in the early 1970's by Codd, a logician, the relational model generated a true revolution in data management. In this simple model data is represented as relations in first order structures and queries as first order logic formulas. It enabled researchers and implementors to separate the logical aspect of the data from its physical implementation. Thirty years of research and development followed, and they led to today's mature and highly performant relational database systems.

The age of the Internet brought new data management applications and challenges. Data is now accessed over the Web, and is available in a variety of formats, including HTML, XML, as well as several application specific data formats. Often data is mixed with free text, and the boundary between data and text is sometimes blurred. The way the data can be retrieved also varies considerably: some instances can be downloaded entirely, others can only be accessed through limited capabilities. To accommodate all forms and kinds of data, the database research community has introduced the "semistructured data model" where data is self-describing, irregular, and graph-like. The new model captures naturally Web data, such as HTML, XML, or other application specific formats.

While researchers mostly agree on a common definition of the semistructured data, there is still a lot of confusion about the logical foundations for representing and querying such data: several practical query languages have been proposed, but their formal foundations and their relationships to logical formalisms are poorly understood. This lack of understanding further prevents us from designing general solutions to

typical data management problems, such as building indexes, optimizing queries, and designing storage structures. To add to the confusion, the structured document community has studied for several years "structured text" and proposed a number of algebraic operators and accompanying index structures to express queries over structured text. This work definitely has relevance to semistructured data, but their connections are still poorly understood. Current work in academia and research institutions is studying the nature of query languages for semistructured data, and proposing index structures, optimization techniques, and storage mechanisms to support those queries.

This seminar aims at bringing together database researchers, logicians, and researchers in structured documents. Furthermore, we would like to invite some people from other communities that are related to the area of semistructured data, like information retrieval, programming languages, and discrete algorithms. Besides the presentation of recent research results by the participants additional goals are:

- to identify the main issues for further foundational research on semistructured data,
- to improve the mutual understanding of the communities involved concerning their respective settings and needs.

24 Ubiquitous Computing

Seminar No. **01371** Report No. **319** Date **09.09.–14.09.2001**
Organizers: Gaetano Boriello, Hans-Werner Gellersen, Friedemann Mattern

Processors are becoming so small and inexpensive that they will be embedded in almost everything. Everyday objects will be infused with computational power, enabling them as information artifacts and smart devices. Most of these new emerging smart devices will be small and therefore highly mobile; some might even be wearable and be worn much as eyeglasses are worn today. Low-cost transceivers will allow to interconnect these devices in spontaneous ways, and to link them into the global information infrastructure. Connected together and exchanging appropriate information, these smart devices will then form powerful systems enabling new emerging functionalities.

When the world is populated with small computing devices that typically do their work in the background, without explicit user intervention, information and computational services will become continuously available, wherever the action is. Moreover, embedded sensors and actuators will enable smart devices and computing to become contextually embedded in real-world situations. This will give rise to situated computer applications that blend with the real tasks people care about instead of introducing computer-centric tasks of high complexity.

Ubiquitous computing therefore induces a paradigm shift in the way we use computers: Instead of bringing the world into the computer (the Virtual Reality paradigm), computational power is now brought to the objects of the physical world. Eventually, the vision of Ubiquitous Computing induces a new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish into the background.

Over the last years, established research communities have begun to relate their fields to the vision of ubiquitous computing, and new communities have emerged to investigate specific perspectives of the development. Researchers begin to consider the enabling technologies and infrastructures required, the new applications and services that may emerge, and the interfaces and human interaction models for ubiquitous computing.

The growing ubiquitous computing community is fed from different classical areas, mostly from within computer science, but also from electrical engineering, material science, product design, and some other disciplines. Hence, insights currently evolve from many different perspectives, but often in parallel and with little interaction.

The Dagstuhl seminar should provide an opportunity to improve this situation by bringing scientists from various relevant disciplines together to jointly discuss the challenges, opportunities, and pertinent research themes of ubiquitous computing. Many participants have their roots in the classical computer science system domains (distributed and mobile computing, networking, architecture, middleware), others will be interested in technologies for smart devices (such as embedded and wearable computing, perception, or knowledge processing), and some will be concerned with application domains and human factors (such as context-aware computing, domestic applications, human- computer interaction, and design).

Abstracts selected by the Dagstuhl News editor:

Interaction in UbiComp: Pirates! and Informative Art

Lars Erik Holmquist, PLAY, Sweden

In this talk, I describe two applications that implement novel interaction techniques for ubiquitous computing. "Pirates!" is a mobile context-aware game developed jointly by the PLAY research group and Nokia Research. The game is played on hand-held computers with proximity sensing. To play the game, users must walk around in the physical play area, in order to engage in combat with other players, find islands with treasure, etc. "Informative Art" uses the visual language of modern art, in particular non-figurative painting, to create dynamic information displays. For instance, a projection display reminiscent of the dutch painter Mondrian's work is in fact a weather display, showing the current weather conditions in six different cities around the world. Together, these projects indicate new ways of using space, mobility and visual media in ubiquitous computing.

25 Algorithmic Aspects of Large and Complex Networks

Seminar No. **01381** Report No. **320** Date **16.09.–21.09.2001**
Organizers: Micah Adler, Friedhelm Meyer auf der Heide, Dorothea Wagner

Large and complex networks play a central role in a variety of today's and future technologies. Communication, information broadcast, and mobile information systems, as well as political and social acting is modeled in terms of such networks. In order to analyze them, the algorithmic aspects of methods to explore them and to derive necessary new information from them have to be understood thoroughly. Scientists from Computer Science, Mathematics, Electrical Engineering and Humanities use these networks, but apply different methods to solve their specific problems.

Algorithmic problems occurring in the design, the analysis, and the application of such networks have been the topic of this seminar. The 33

participants covered a wide area of aspects of this research field, e. g., graph theoretical and combinatorial foundations, graph algorithms, design and analysis of networks via optimization and approximation techniques, reliability and security aspects, and applications in parallel and distributed computing and social networks.

Abstracts selected by the Dagstuhl News editor:

Network Design: Modeling and Solving in the Airline Context

Georg Kliewer (joint work with Achim Koberstein)

The airline network design problem appears in the long-term planning phase of an airline. The goal is to decide for a given flight network which arcs can be eliminated or added. The proposed solution must take the passenger flow in the network into account and also the aircraft flow of the airline. For each arc a binary design decision variable is defined. The passenger flow is modeled as a minimum cost multicommodity flow. The model for the aircraft flow considers different aircraft types, balancing conditions, aircraft availability, etc. We work with path-based multicommodity flow models because of the fact that a passenger travel path is strictly constrained, e.g., the path length cannot be too long. Our experiments include CPLEX-based solution procedures and also a metaheuristic simulated annealing algorithm. We discuss results obtained for the flight network of Lufthansa. A future work direction is to use the network design decisions for the airline alliance flight network planning.

26 Specification and Analysis of Secure Cryptographic Protocols

Seminar No. **01391** Report No. **321** Date **23.09.–28.09.2001**
Organizers: David Basin, Grit Denker, Gavin Lowe, Jon Millen

Cryptographic protocols are the cornerstone of secure electronic communication, banking, and commerce. By providing functions like key management in distributed systems, individual and group authentication, anonymity, fair exchange, and policy negotiation, they support a

spectrum of secure online activities such as financial account transactions, distributed sealed-bid auctions and their escrow services, voting, distributed and federated database access, and virtual private networks.

Designing cryptographic protocols is difficult. Cryptographic protocols are vulnerable to message modification attacks and it is surprisingly difficult to get even small protocols right. Moreover, their complexity is steadily increasing and it is nontrivial to compose or extend smaller protocols to more complex ones.

Formal methods have proven helpful for both cryptographic protocol design and analysis. The use of formal languages, including state machines, epistemic logics, and process algebras, supports the rigorous formalization of protocol models and their properties. Moreover, they provide a basis for using tools such as model checkers and theorem-provers to prove protocols correct or uncover security flaws.

The goal of this seminar is to bring together experts from the formal methods and security communities to study and compare existing modeling and analysis techniques and tools for cryptographic protocols, focusing on the following topics:

1. identifying and formalizing appropriate and practical security goals and security protocol analysis techniques,

2. classifying and comparing existing modeling techniques and formalisms, and

3. comparing existing tools, identifying the most urgent needs to integrate formal methods into real world protocol design, and suggesting future directions for tool design.

There are many questions and unresolved issues associated with each of these topics.

Regarding (1), security goals include message and entity authentication, secrecy, anonymity, integrity, non-repudiation, fair exchange, agreement, and denial of service resistance. There is currently no commonly accepted formal model of all of these and it is not even clear if this list is complete. Additional questions include: Which tool or analysis technique best supports which kind of goals? What is the relationship between formalizations of goals in different frameworks? Which protocol technique can handle "weak secrets," i.e., secrets that can be revealed

by guessing? What security goals are essential in group communication applications? In this seminar we aim to identify commonly understood security goals and security protocol techniques and means to link goals with appropriate verification tools.

Regarding (2), currently many different techniques are used for modeling, e.g., event-based approaches using knowledge and belief logic abstractions, agent-based approaches modeling protocol processes using multiset rewriting, process algebra based approaches, and the strand space approach. How do these models differ? Natural candidates for a classification scheme are: synchronous versus asynchronous communication, complexity, decidability, practicability, which class of cryptographic protocols can be modeled (point-to-point, group communication, etc.), which cryptographic and other computations are supported, which analysis techniques are supported, and what is the scope, extensibility, and reusability of the modeling formalism. Heuristics and transformations play a role here too: how can protocols and their models be safely transformed to bring them within the reach of current tools (e.g., reduced to small finite-state programs that can be model checked)?

Regarding (3), more sophisticated tools are required if protocol designers are to use them for real world problems. To what extent can tools scale that are based on (potentially infinite) state enumeration, finite state model checking, or automated (or interactive) theorem proving? In automated approaches, heuristics often play an important role in incorporating domain knowledge about particular protocols into the search process. Tool design issues include: How can the performance of heuristics be enhanced by exploiting knowledge about the protocol, message formats or attacker behavior in a rigorous way? Can heuristics be designed in a general, reusable style but still amenable to optimization? Finally, how can model checkers and theorem provers fruitfully interact with each other (e.g., the use of theorem provers to verify abstractions for model checking)?

Participants expressed interest in archiving the slides of presentations. There is now a web-page,
`http://www.informatik.uni-freiburg.de/~accorsi/dagstuhl/`, that contains this information. Moreover, current plans are for producing a "state-of-the-art" based, in part, on the workshop. Future plans also include the design of a web-site providing a centralized bibliography service on verification of security protocols. This web-site, which is currently in experimental phase, can be found at

http://www.informatik.uni-freiburg.de/~accorsi/protsecweb/.

Abstracts selected by the Dagstuhl News editor:

A flaw in a denial-of-service resistant protocol

Tuomas Aura

The possibility of formally specifying and analyzing DoS attacks and resistance was contemplated after Cathy Meadows's talk on DoS. Since there is a shortage of concrete examples, I presented a real but difficult-to-detect flaw in an early version of a DoS-resistant protocol.

In the protocol, the client must solve a cryptographic puzzle before the server commits its memory and processing power to the authentication. In order to exhaust the server's resources, an attacker would need to solve large numbers of such puzzles, which would be too expensive. The problem is that while the protocol protects the server against flooding attacks, the attacker can deny service for individual clients by solving the puzzles for them and submitting the solutions to the server before the client does. The server will remember the solved puzzles and ignore duplicate solutions sent later by the real client. The problem is solved by a minor modification of the puzzle. The flaw would probably have been spotted earlier in the design process, had there been a set of well-defined DoS-resistance requirements.

Lazy analysis of security protocols

David Basin

We present an approach to modeling security protocols using lazy data types in a higher-order functional programming language. Our approach supports the formalization of protocol models in a natural and high-level way, and the automated analysis of safety properties using infinite-state model checking, where the model is explicitly constructed in a demand-driven manner. We also discuss recent extensions of this work and the construction of a general purpose protocol analysis tool.

27 Proof Theory in Computer Science

Seminar No. **01411** Report No. **322** Date **07.10.–12.10.2001**
Organizers: Reinhard Kahle, Peter Schröder-Heister, Robert F. Stärk

Proof theory has long been established as a basic discipline of mathematical logic. In recent years it has become increasingly relevant to computer science. The deductive apparatus provided by proof theory has proved to be useful both for metatheoretical purposes and for practical applications in various fields of computer science.

The aim of this conference is to assess which role proof theory is already playing in computer science, and which role it might play in further developments. Is proof theory going to be the most preferential approach to the logical foundations of computer science? Does is provide viable alternatives in areas where model-theoretic approaches are predominant?

A central focus of the conference may be captured by the slogan *logics for programs*, i.e. the proof theoretic approach in dealing with design, development and application of programming languages.

Major divisions of PTCS are the following (but this list is not intended to be exclusive):

- The *proofs as programs* paradigm in general
- Typed and untyped systems related to functional programming
- Proof-theoretic approaches to logic programming
- Proof-theoretic ways of dealing with computational complexity
- Proof-theoretic semantics of languages for specification and programming
- Foundational issues

Proof theory is not a uniform subject at all. The list of invited participants includes researchers from different research paradigms. In particular, we are inviting both proof theorists in the more traditional "theoretical" or "mathematical" sense, and computer scientists using proof theoretic tools in the area of deduction.

Abstracts selected by the Dagstuhl News editor:

The Birth of Combinators and Lambda

Roger Hindley (joint work with Felice Cardone)

Despite a very natural tendency for combinatory logic to be seen as a modification of λ-calculus, combinators were actually invented in 1920, 8 years before λ. Further, they were invented at least twice : by Schönfinkel in Göttingen in 1920, by von Neumann in Göttingen (perhaps independently ?) in 1925, and by Curry in Harvard in 1927. They gave rise to two results : eliminability of bound variables (Schönfinkel and Curry), and finite axiomatizability of set theory (von Neumann). Then in about 1928, the λ-calculus was invented by Church as part of a general system intended to be an axiomatic basis for all of mathematics. This system was inconsistent, but its core was the pure λ-calculus, which gave him his proof of undecidability of the Entscheidungsproblem. This talk is based on material from an article by Cardone and Hindley on the history of lambda and combinators, for a book on the history of mathematical logic, to be published by North-Holland Elsevier.

Implicit characterizations

Isabel Oitavem (contains joint work with S. Bellantoni)

Several machine independent approaches to relevant classes of computational complexity have been developed. All of them lead to characterizations of classes of computational complexity where the machine formulation and resources are implicit – implicit characterizations. We work in a free algebra context. That allows us to obtain implicit characterizations of classes as different as Ptime, Lspace and NC just by changing the starting free algebra. By doing so, we give a uniform approach to classes which result from processes as different as deterministic and parallel computations with time, space or time and space constraints.

The last part of this talk is devoted to Pspace. We describe and discuss implicit characterizations of Pspace.

28 Integration of Algebra and Geometry Software Systems

Seminar No. **01421** Report No. **323** Date **14.10.–19.10.2001**
Organizers: Michael Joswig, Nobuki Takayama

In many fields of modern mathematics specialized scientific software becomes increasingly important. Therefore, tremendous effort is taken by numerous groups all over the world to develop appropriate solutions. Topics include commutative algebra, D-modules, group and number theory, as well as algebraic, computational, discrete, and differential geometry.

With growing complexity of these software systems, the designers more than ever feel compelled to include functionality which might require techniques other than the roots of the respective system. Those techniques can often be found in other branches of mathematics. Instead of implementing new software individually it is preferable to make use of already existing stable software written by experts in their field. This raises the question for interfaces between software components. Several frameworks have been suggested and are still being discussed.

The seminar covered the following topics:

- Algorithms which require components from both geometry and algebra.
- Abilities and limitations of existing software systems from algebra and geometry.
- General purpose interfaces for mathematical software.
- Visualization components.
- XML based techniques for the presentation and the exchange of mathematical objects.

To ourselves, as the organizers, the undertaking of this workshop looked somewhat ambitious at the beginning. The participants came from quite diverse areas and we were a little afraid that there might be too little common ground. So we were relieved to see the participants

enter very lively discussions in a friendly and open minded way from the first day. We are sure that the wonderful atmosphere of Schloss Dagstuhl helped a lot.

We hope that some of this spirit will still be visible in a forthcoming book on the workshop's subject to be published by Springer.

Abstracts selected by the Dagstuhl News editor:

OpenMath Tools

Arjeh M. Cohen (TU of Eindhoven)

As a result of the OpenMath ESPRIT project, there is an OpenMath standard for the representation of mathematical objects, there are Content Dictionaries (approved by the OpenMath Society) for objects corresponding to MathML objects, and there are tools for manufacturing phrasebooks. We have explained that an OpenMath expression is a tree whose leaves can be four kinds of "constant", variables, or symbols. The latter are defined in Content Dictionaries (CD). Each symbol is defined in a unique CD.

The official CDs can be found at `http://www.openmath.org`. But there are quite a few more. Some are experimental, but many others are meant for communication between a few applications, programs, or agents, and far from being as public as the official ones.

A phrasebook is understood to be the codec (taking care of translations from and to the application) together with the program that controls the communication. It is emphasized that the phrasebook needs to specify which CDs it uses and which actions it will perform on the OpenMath objects received.

There are three libraries for constructing phrasebooks. The one constructed at Eindhoven can be found at `http://crystal.win.tue.nl/download`, and contains several examples of existing phrasebooks, notably for Mathematica, COQ, and GAP. A brief discussion of the use of OpenMath for mathematical interaction on the Web, and requirements (such as a query language and brokerage) ended the talk.

Object-oriented Design and Mathematics

Marc Conrad (Southampton Institute)

Object oriented programming has well been established as a software design paradigm in the last years and is widely used in commercial software projects. However — in the context of mathematics it is rarely used, usually (if at all) in connection with a concrete programming language as C++ or Java. In contrast, we take here a language independent view, and show that there is a strong correspondence between mathematical structures and object oriented concepts.

For instance, inheritance is similar to the specialization in mathematics (*"A field is a ring with ..."*), overriding of deferred methods coincides with the step from an abstract concept to a concrete example (*"Define* x^2 *as* $x \cdot x$ *and compute* 3^2*"*), and polymorphic behavior corresponds with defining concepts on an abstract level (*"An elliptic curve over a field."*).

The systematic assigning of responsibilities to objects is shown on the example of elliptic curves.
(Joint work with Susanne Schmitt, Saarbrücken)

29 Plan-based Control of Robotic Agents

Seminar No. **01431** Report No. **324** Date **21.10.–26.10.2001**
Organizers: Michael Beetz, Joachim Hertzberg, Malik Ghallab, Martha Pollack

In the Seminar, we brougth together a team of 34 leading researchers who are researching different aspects of plan-based control such as flexible and reliable execution of plans, execution time plan formation and revision, and automatic learning of robot plans. They discussed issues in plan-based control and worked towards a comprehensive framework for plan-based control of robotic agents. In the repesentations and discussions we were particularly addressing the following issues, both from the methodological and the application side.

- **Flexible and reliable execution of plan-based control.** What is the right representation and expressivity of plans for autonomous robot control? Should the plan language support flexible and reliable execution? What kind of plan management operations other

than plan formation should the plan representation facilitate? How can the plan representation be grounded into the sensory and actuation capabilities of the robot?

- **Realization of runtime plan management and the integration of plan management into the overall control.** How can we accomplish the feasibility of plan management operations? Should plan management be incorporated at a plan layer of a hybrid robot control architecture? Are there other means for integrating plan management into the overall control?

- **Automatic learning of plan schemata and plan revision knowledge.** How can the robot automatically chunk its continuous behavior and learn complex behavior structures? How can the robot automatically adapt to specific environments and tasks? How can the robot learn or acquire the plan schemata and planning knowledge needed for execution time plan management?

- **Formal models of plan-based control, plan formation, and plan revision.** How can we obtain realistic formal models of plan-based control? Can formal languages be used to define an abstract description layer for robot controller design? Do formalizations help clarifiy robot/human and robot/robot interaction, high-level task-oriented command, and tele-manipulation interface languages?

- **Applications of plan-based robot controllers.** What are the challenge applications for plan-based control of robotic agents? Are there empirical results on the use of plan-based versus behavior-based robot controllers? Should we try to collect benchmarks for plan-based robot control?

The seminar covered the topics listed above by a program of long talks, technical talks and panel discussions. Selected contributions to the Seminar are currently under review and will be published as a book in the Springer Publisher's Lecture Notes in Artificial Intelligence series.

In the Seminar we have collaborated with PLANET's (the EU Network of Excellence in AI Planning and Scheduling), in particular the PLANET technical coordination unit (TCU) on Robot Action Planning. With this collaboration we could exploit synergies between the two activities. PLANET has provided us with an excellent infra structure and

support for European researchers in the field. On the other hand, the Seminar has provided valuable input to PLANET's road map for research in robot planning. Nationally, the seminar has cooperated with the DFG *Schwerpunktprogramm Kooperierende Teams mobiler Roboter in dynamischen Umgebungen* (Cooperating Teams of Robots in Dynamic Environments). It was an explicit aim of the seminar to bring participants of these two activities together and, moreover, to deepen the contact with researchers from the U.S. working in the field.

The seminar has fueled several initiatives for research project proposals and multi lateral cooperations between participants of the seminar. In one of them we intend to make a project proposal that comprises several European research groups to develop a common testbed for plan-based control of autonomous robots. This intended project should then be followed up in a broader context and in the form of a basic research initiative within the next ESPRIT framework.

Other results of this seminar include a better, deeper understanding of issues and methodologies in plan-based control of robotic agents as well as insights into relevant real-world applications of this technology. The organizers also intend to write an article about the state of the field based on the Seminar and submit it to a magazine.

Many participants have expressed that they found the seminar highly interesting and stimulating. The typical approach of Dagstuhl seminars to bring together for ample exchange researchers from different communities has once more proven its charm, where mobile robotics, artificial intelligence and agent technology were the communities most prominently present. A follow-up seminar under the same title will be held in 2003, giving those who couldn't make it in 2001 the opportunity to meet those who have expressed their firm intention to come back next time.

Abstracts selected by the Dagstuhl News editor:

Strength in Numbers: A Team of Robotic Agents for Surveillance

Maria Gini

This talk presents the hardware and software components of a robotic team designed for security and surveillance applications. The team consists of two types of robotic agents. The first type is a larger, heavy-duty

robotic platform, called the "ranger." Rangers are used to transport, deploy, and supervise a number of small, mobile sensor platforms called "scouts," the second type of robotic agent. In an example scenario, the scouts are deployed into an office/lab environment, navigate towards dark areas, and position themselves to detect moving objects using their cameras. A ranger communicates with each of the scouts and determines whether there are objects of potential interest within the observed area. The paper includes experimental results for individual scout and ranger-scout activities.

30 Exploration of Large State Spaces

Seminar No. **01451** Report No. **326** Date **04.11-09.11.2001**
Organizers: Tom L. Dean, Bernhard Nebel, Moshe Y. Vardi

Description

The goal of this workshop is to bring together researchers working on state-exploration methods in artifical intelligence (AI) and in automated verification (AV). The idea of organizing such a workshop came during the DIMACS/IRCS Tutorial and Workshop on Logic and Cognitive Science, which was held in April 1999 at the University of Philadelphia. We realized there that state-exploration is a major issue in computer-aided verification and in artificial intelligence.

Automated Verification

Automated verification provides a new approach to validating the correct behavior of software and hardware designs. In traditional design validation, "confidence" in the design is the result of running a large number of test cases through the design. Automated verification, in contrast, uses mathematical techniques to check the entire state space of the design for conformance to some specified behavior. Thus, while simulation is open-ended and fraught with uncertainty, formal verification is definitive and eliminates uncertainty. Over the last few years, automated verification tools, such as model checkers, have shown their ability to provide thorough analysis of reasonably complex designs. Companies such as AT&T,

Cadence, Fujitsu, HP, IBM, Intel, Motorola, NEC, SGI, Siemens, and Sun are using model checkers increasingly on their own designs to ensure outstanding product quality. Unfortunately, model checking suffers from a fundamental problem known as state-explosion: the ability to handle only systems with limited-size state spaces. This explosion arises mainly because the transition system analyzed describes the global behavior of the system. In a system comprised of multiple components, the global state space is a cross product of the individual component state spaces. Even a system containing only small components can therefore yield a large global state-space. For a modern hardware and software, the global state space is prohibitively large. This poses a serious challenge to extant model checkers. Combating state-explosion is therefore a complex computational problem of critical importance.

Markov Decision Processes

Work on Markov decision processes (MDPs) dates back to the 1950's and there is a large literature stemming primarily from the fields of operations research and adaptive control. In the 1980's there was renewed interest in MDPs coming from work in automated planning in artificial intelligence and the work on binary decision diagrams from the verification community. Dean and Kanazawa [1989] and Tatman and Shachter [1990] introduced the idea of structured Markov processes and suggested how they might be used for representing and solving planning and control problems with very large state and action spaces. In subsequent years, a great deal of progress was made exploring structured versions of earlier, unstructured algorithms from the operations research and adaptive control communities.

True MDPs, i.e., problems in which the current state of the system is completely observable to the decision maker, are rare in practice and hence the partially observable variant (POMDP) is of great importance. Recently there has been a resurgence of interest in POMDPs, partially spurred by the success of Cassandra, Kaelbling and Littman, and a host of new algorithms have been developed, including variational methods which open up the possibility of solving a wide range of problems. Variational statistical methods can in some cases reduce the need for state exploration using a combination of sampling techniques and reformulation in terms of a (continuous) parameterized space of actions.

More recently researchers have begun using the basic technology for model checking including binary decision diagrams (BDDs), various algebraic extensions, and quantified variants that make use of modal logics of time to tackle a wider class of MDPs and their partially observable counterparts. In all cases, the need for exploring very large state and action spaces is dealt with by identifying and then exploiting structure in the combinatorial space of states and actions.

AI Planning

Planning is a sub-field of AI which is concerned with the generation of a rational course of action given a declarative specification of the environment, the goals, and the possible actions. In the case of "classical planning," one usually makes the simplifying assumptions that everything (that is relevant) is observable, that all actions are deterministic, and that the only change in the world is caused by actions of the agent. While this seems like a trivialization of the MDP problem, the problem is still computationally very hard because of the large state space one has to explore. The last five years brought considerable progress on the algorithmic side. In particular, the planning-graph approach by Blum and Furst [1995], which outperformed any existing planning system then, led to a flurry of further developments, such as planners that reduce the planning problem to a sequence of propositional satisfiability problems [Kautz and Selman 1996] or nonmonotonic reasoning problems [Dimopoulos at al 1997]. Type inference algorithms, the derivations of invariants and other techniques were used to prune the search space and to speed up planning, and OBDDs were used to code sets of states in a compact way. All these developments led to an impressive performance of AI planning systems as demonstrated at the first international planning competition in 1997 (at AIPS'97). Just to give an example, Kautz and Selman's BLACKBOX planner produced a 100-action plan in a world with approx. 10^{16} states in a few minutes. This has to be compared with the state of the art in the beginning of the nineties, when a 10-action plan needed a few hours to be generated. Currently, the main focus of research is in extending the existing techniques to handle more expressive planning formalism (e.g., including resources), in improving the core search algorithms, and in exploiting all types of knowledge that can be derived from the problem specification in order to prune the search space.

31 Computability and Complexity in Analysis

Seminar No. **01461** Report No. **327** Date **11.11.–16.11.2001**
Organizers: Vasco Brattka, Peter Hertling, Mariko Yasugi, Ning Zhong

This meeting was the third Dagstuhl seminar which was concerned with the theory of computability and complexity over the real numbers. This theory, which is built on the Turing machine model, was initiated by Turing, Grzegorczyk, Lacombe, Banach and Mazur, and has seen a rapid growth in recent years. Recent monographs are by Pour-El/Richards, Ko, and Weihrauch.

Computability theory and complexity theory are two central areas of research in theoretical computer science. Until recently, most work in these areas concentrated on problems over discrete structures. In the last years, though, there has been an enormous growth of computability theory and complexity theory over the real numbers and other continuous structures. One of the reasons for this phenomenon is that more and more practical computation problems over the real numbers are being dealt with by computer scientists, for example, in computational geometry, in the modelling of dynamical and hybrid systems, but also in classical problems from numerical mathematics. The scientists working on these questions come from different fields, such as theoretical computer science, domain theory, logic, constructive mathematics, computer arithmetic, numerical mathematics, analysis etc.

This Dagstuhl seminar provided a unique opportunity for 46 participants from such diverse areas and 12 different countries to meet and exchange ideas and knowledge. One of the topics of interest was foundational work concerning the various models and approaches for defining or describing computability and complexity over the real numbers. We gained new insights into the computability-theoretic side of various computational questions from physics as well as from other fields involving computations over the real numbers. The 39 talks also covered fields closely connected to computable analysis such as recursion theory, algorithmic information theory, constructive mathematics, realizability theory, domain theory, proof theory, complexity theory and interval analysis. Last but not least, new implementations of exact real arithmetic and further developments of already existing software packages have been

discussed.

32 Synchronous Languages

Seminar No. **01491** Report No. **328** Date **02.12.–07.12.2001**
Organizers: Willem-Paul de Roever, Nicolas Halbwachs, Gérard Berry, Klaus Winkelmann

Synchronous Languages: An Introduction for Laymen

Synchronous languages provide a solution to the following problem:

> *"What is the highest level of specification possible for real-time embedded systems?"*

Previous studies had demonstrated that, at such a high level, the notion of time should not be quantitative, e.g., no mention should be made that an action lasted 1.5 μsec. This is because real-time embedded systems are, in general, too complicated to also specify the time needed by their components; mentioning quantitative time would lead to loss of the overall picture.

Independently, 4 teams (3 French ones and 1 Israeli one) found solutions to this problem in 1980–1985, leading to the synchronous languages LUSTRE, ESTEREL, SIGNAL, and the semi-synchronous visual language StateCharts, supported by the tool StateMate.

Their underlying design assumption is that internal (re)actions take no time to execute, in comparison with the external stimuli requesting them ("Berry's synchrony hypothesis"). Although this is in its pure form unimplementable, it may very well be the case that internal (re)actions always take less time to execute than passes between two successive external stimuli; this must be proved for each particular application.

Also, these languages execute concurrent actions synchronously in multi-steps, and have the possibility to test for the absence of an internal action. Simple as these design decisions may seem, they have led to languages with enough expressive power to specify the control of jet engines, nuclear reactors and on-board software. Also, due to their simplicity, their compilers could be proved correct. As a result, the compiled

code only contains errors due to wrongly expressed specifications at the source level.

In case of embedded software for the line of aircraft produced by AIRBUS, and control software for nuclear power stations (produced by Schneider Inc.), the utilization of synchronous languages led to a decrease of more than 95 % of errors per 10.000 lines of code.

With their design criteria and implementation techniques now firmly grounded in theory, two main problems remain:

1. How to combine asynchronous (re)actions with the synchronous approach into a unified framework? For, obviously, e.g., in software for airplanes, asynchronous delay is only natural.

2. How can synchronous languages be adapted to the description of hardware, i.e., synchronous circuits. For, due to the fact that the amount of tested runs in modern VSLI chips approaches rapidly zero, when compared with their overall capabilities, new languages and tools, with much faster execution speeds, are needed to raise the possibility of testing Pentium-class VLSI chips to acceptable levels. As the use of ESTEREL as hardware-description language has demonstrated, it is feasable to do this using the synchronous-languages approach.

Apart from presenting results on these two main issues, the seminar focussed as third issue on visual methods for specifying synchronous languages, an approach pioneered by the StateMate system designed by David Harel, and now incorporated also in the two mainstream French visual specification tools, based on synchronous languages: SCADE (combining LUSTRE with elements of StateCharts), and Esterel-Studio (based on a visual representation of ESTEREL, called Synch-Charts).

This series of seminars constitutes the only yearly meeting place for the researchers in this exciting field.

Kiel, February 27, 2002 W.-P. de Roever