Egon Börger, Yuri Gurevich,
Hans Kleine-Büning, M.M. Richter (editors):

Computer Science Logic

Dagstuhl-Seminar-Report; 40
13.07.-17.07.92 (9229)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

Computer Science Logic

13.-17.7.1992

Organizers:

EGON BÖRGER (Universitá di Pisa, Italia)
YURI GUREVICH (University of Michigan, Ann Arbor, USA)
HANS KLEINE BÜNING (Universität Paderborn, Germany)
MICHAEL M. RICHTER (Universität Kaiserslautern, Germany)

Research in the area between computer science and logic is fast growing and already shows a tendency to split into various subareas. The aim of this workshop was to bring together eminent researchers of the most active and internationally recognized research lines of computer science logic in order to discuss and critically reflect upon the fundamental common problems, concepts and tools. 42 top scientists and promising young researchers accepted the invitation to participate in the challenging experience. They came from 14 countries, 1/6 from USA, 2/6 from Germany, the remaining half from other European countries, including East Europe, and Israel.

Talks covered semantics, specification and correctness proofs, complexity of logic algorithms, finite model theory, logic programming, lambda-calculus and functional programming, rewriting, non-classical reasoning (in particular non-monotonic, temporal, epistemic and many-valued logics), linear logic and proof theory. Many new methods and ideas were presented which open new lines of promising research in computer science logic. Critical discussions attempted to identify and examine the underlying fundamental problems.

The atmosphere was very friendly, but the discussions were most lively and participants did not hold back their critical remarks. The discussions turned many lectures – scheduled for daily morning and afternoon sessions – into long disputations; this is a Dagstuhl effect that cannot be overstimated. During the breaks and till late in the night, participants also gathered in smaller groups for continuing discussions, communicating new results and exchanging ideas. During the week, numerous participants worked together and some were inspired to start new research projects; this is one of the most fruitful outcomes of the workshop.

The success of the workshop exceeded our expectations. The participants expressed high appreciation of this gathering and praised the extraordinary Dagstuhl atmosphere which made this success possible.

Summing up intensive discussions over the last years in the Computer Science Logic community, on the evening of July 14, 37 of the participants from 14 European countries, USA and Israel founded the European Association for Computer Science Logic. This association which grew out from the CSL workshop series, created by three of us in 1987 and run since then on an annual basis, aims at promoting research and international cooperation in the area between logic and computer science.

As organizers of the Dagstuhl workshop on Computer Science Logic and on behalf of its participants we want to thank the institute and its staff, both in Saarbrücken and in Dagstuhl, for the excellent work they did to make it all run smoothly in an efficient but always pleasant and friendly manner. They did much more than just their job; their inspiring dedication to 'their' institute makes Dagstuhl a perfect place. We also thank Dean Rosenzweig, who has edited this report.

Firstorder properties of random substructures

MIKLOS AJTAI

Assume that a structure is given on the finite universe A and f is a random function mapping the set F into A. The map f induces a structure on F in a natural way.

We show that if F has n elements and φ is a not too large firstorder formula, then with a probablity of at least $1 - n^{-\epsilon}$ it is enough to randomize f on $n - n^{\epsilon}$ points to get the value of φ on F. Moreover, with a probability exponentially close to 1, after this randomization φ collapses to a $\sum_2$-formula.

The $\forall\exists$–Theory of the Polynomial Many–One Degrees is Decidable

KLAUS AMBOS–SPIES
(Joint work with M.Lorman)

Ambos–Spies and Nies (STACS '92) have shown that the theory of the polynomial–time many–one (p–m) degrees of recursive sets is undecidable. This leads to the question on which quantifier level undecidability starts to occur.

Here we give a decision procedure for the $\forall\exists$–theory. This procedure does not only work for the domain of all recursive sets, but can be adapted to many of the standard complexity classes. E.g. we obtain a decision procedure for the $\forall\exists$–theory of the p–m–degrees of EXPTIME sets.

Lazy Functional Logic Programming with Disequality Constraints

MARIO RODRIGUEZ ARTALEJO

We use an approach to the combination of lazy functional programming, logic programming and constraint programing, based on a general scheme CFLP(X) for Constraint Functional Logic Programming over a constraint structure X (a continuous algebra) which can be supplied as a parameter. CFLP has been recently proposed in [Lopez Fraguas 92] as a natural generalization of the well known scheme CLP defined by Jaffar and Lassez for Constraint Logic Programming.

We first report on the general properties of CFLP, which essentially ensure the existence of least models for consistent programs, as well as the soundness and completeness of an operational semantics. Models must be persistent extensions of the base constraint structure, while the operational semantics is based on lazy constrained narrowing. "Laziness", in this context, means that the goal is reduced only by narrowing at "demanded" positions. What positions are regarded as demanded depends on semantic information given by the least model. Hence, these general results provide no effective way

of implementing a goal reduction strategy. This problem has to be faced for each particular instance CFLP(X) of the CFLP scheme, on the basis of X-dependent goal reduction methods.

Next, we define a particular instance CFLP(H) of CFLP, using a constraint structure H which supplies finite and infinite trees built from constructors, equality and disequality constraints, and some other primitive operations. In contrast to the constraint structure underlying Prolog II, infinite trees in H may be non rational. We show by examples that the resulting programming language supports interesting combinations of constraint logic programming and programming with lazy functions. We introduce a paricular presentation of CFLP(H)-programs (the uniform CFLP(H)-programs) which allows to specify a more efficient operational semantics for constrained lazy narrowing, by taking advantage of specific features of H. The H-specific formulation of the operational semantics for uniform programs includes an effective strategy for selecting demanded positions. A more machine-oriented approach towards the implementation of a language essentially equivalent to CFLP(H) has been also undertaken [Kuchen et al. 92].

We conclude that the scheme CFLP is a meaningful extension of CLP, which stimulates new views of declarative constraint programming and admits sensible instances. More work is still needed to understand the general properties of the scheme and to develop practical implementations of useful instances.

References

[Lopez Fraguas 92] F.J.Lopez Fraguas: *A General Scheme for Constraint Functional Logic Programming*, to appear in Procs. ALP'92.

[Kuchen et al. 92] H.Kuchen, F.J.Lopez Fraguas, J.J.Moreno Navarro, M.Rodriguez Artalejo: *Implementing a Lazy Functional Logic Language with Disequality Constraints*, to appear in Procs. ICLP'92.

On Gödel's Theorems on Lengths of Proofs

SAMUEL R. BUSS

We write $\vdash_k$ to denote provability by a proof of $\leq k$ lines and $\vdash^k$ to denote provability by a proof of $\leq k$ symbols. Z_i denotes $(i+1)$-st order arithmetic; the function symbols may include $+$ and $\cdot$ as well as (optionally) function symbols for all primitive recursive functions. Z_i must be formalized in a 'Hilbert-style' calculus with schematic rules and possible with all tautologies as axioms (a 'weakly schematic' system).

The first part of our talk discusses Gödel's 1936 paper on proof speedup in higher-order systems of arithmetic. Various authors have proven analogues of Gödel's theorem with proof length measured by number of symbols; Parikh and Krajíček gave a proof for the case where successor is the only function symbol. We give the first publicly known proof of the exact theorem stated by Gödel, to wit, we show:

Theorem: Let $i \geq 1$. There is an infinite family $\mathcal{F}$ of Π^0_1-sentences and an integer k so that, for all $\phi \in \mathcal{F}$, $Z_{i+1} \vdash_k \phi$ and $Z_i \vdash \phi$ but so that there is no uniform upper bound on the number of lines in Z_i proofs of ϕ. Thus Z_{i+1} has unbounded speedup over Z_i.

The second part of our talk concerns a recently discovered 1956 letter of Gödel to von Neumann in which Gödel discusses the feasiblity of deciding whether a given formula has a proof of $\leq n$ symbols in first-order logic. We speculate about Gödel's motivations, noting especially the fact that "$\vdash^n \phi$?" is NP-complete (n is to be given in unary notation).

We prove the following theorems:

Theorem Also for propositional logic, the decision problem "$\vdash^n \phi$?" is NP-complete.

This theorem is somewhat surprising since the set of tautologies is coNP-complete.

Theorem Given a fixed alphabet A and fixed number k of tapes, there is a constant ϵ such that any deterministic Turing machine with alphabet A and k tapes which decides "$\vdash^n \phi$?" takes time $> \epsilon \cdot n$ infinitely often. (Even with ϕ required to have $O(\log n)$ symbols.)

This last theorem was made as a claim in Gödel's 1956 letter. In addition, we prove a conjecture of S. Cook's that, if the Turing machine in the final theorem is nondeterministic, then it can not have runtime $o(n/\log n)$.

Embedding of Combinatory Algebras into Themselves

CORRADO BÖHM

From the computational point of view Combinatory Logic and λ-calculus act at the same time as high-order functional programming interpreters and as low-level abstract machines. Combinatory congruences and λ-calculus convertibility classes are too refined to model usual equivalence classes of programs.

A motivation for the present just beginning research is to develop a new notion of equivalence between combinators. Two combinators are equivalent on some predetermined combinator domain if the result of applying both combinators to the same arbitrary element of the domain are congruent. The domains we are interested to, have the shape $F\,x$ or $E\,x$, where E is a left-inverse of F, and x is any combinatory term (variables are also admitted).

An attractive way to study these new equivalences is offered by the introduction of privileged internal models of combinatory algebras, essentially represented by two combinators, the combinator Ap_F corresponding to the application combinator inside the model and a left-invertible combinator F defining the model domain $F\,x$. The model is just obtained embedding the combinatory algebra into itself.

To any given F it corresponds an infinity of mutually isomorph models (Ap_F, F), where all Ap_F are mutually equivalent. With the help of an idempotent combinator $R_F E$, associated to the choice of F and E, we can express combinators possibly transforming an Ap_F into a new one, and having interesting properties.

All these results, if applied to two classic relations respectively discovered by Curry and by Church, and defining implicitly two privileged internal models, give some intuition of the underlying structure.

A new specification and correctness proof for the WAM

EGON BÖRGER

In joint work with Dean Rosenzweig we provide a new proof for our theorem on the correctness of (a formal model of) the Warren Abstract Machine. Starting from an abstract Prolog tree model which is close to programmer's intuition, we derive the WAM methodically by stepwise refinement of Prolog models, proving correctness and completeness for each refinement step. Along the way we explicitely formulate, as proof assumptions, a set of natural conditions for a compiler to be correct, thus making our proof applicable to a whole class of compilers.

The proof method, which improves considerably our previous work in Springer LNCS 533 and 592, provides a rigorous mathematical framework for the study of Prolog compilation techniques.It can be applied in a natural way to extensions and variants of Prolog and related WAMs allowing for parallelism, constraint handling, types, functional components. We reach full mathematical rigour, without heavy methodological overhead, by using Gurevich's notion of evolving algebras.

Minimal Space Requirements for Resolution

HANS KLEINE BÜNING

It is well known that resolution is complete and sound for formulas in conjunctive normal form. Haken has shown that there is an infinite family of propositional formulas for which the shortest resolution refutation requires superpolynomially many resolution steps. At least as important as the length of refutations seems to be the space we need for the answer whether the formula is satisfiable. There are several non- deterministic and deterministic satisfiability algorithms solving the problem in linear or quadratic space. Take for example one of the Davis–Putnam algorithms or evaluate the formula by the sequence of possible truth assignments.

In case of resolution we can not store each clause of a refutation, because refutations of superpolynomially length exists. But we show that we do not have to store each deduced clause in order to obtain the empty clause. We prove that for each unsatisfiable formula F there is a sequence of resolution steps leading to the empty clause, such that memory with at most $2length(F)$ places for clauses suffices.

That means resolution requires at most quadratic space in the length of the initial formula for nondeterministic applications of resolution and there exists a polynomial upper bound for a deterministic resolution algorithm.

Cutting planes and constant depth Frege proofs

PETER CLOTE

The cutting planes refutation system for propositional logic is an extension of resolution and is based on showing the non-existence of solutions for families of integer linear inequalities. We define a modified system of cutting planes with limited extension and show that this system can polynomially simulate constant depth Frege proof systems. Our principal tool to establish this result is an effective version of cut elimination for modified cutting planes with limited extension. Thus, within a polynomial factor, one can simulate classical propositional logic proofs using modus ponens by refutation-style proofs, provided the formula depth is bounded by a constant. Since there are polynomial size cutting planes proofs for many elementary combinatorial principles (pigeonhole principle, Ramsey's theorem), we propose propositional versions of the Paris–Harrington theorem, Kanamori–McAloon theorem, and variants as possible candidates for combinatorial tautologies which may require exponential size cutting planes and Frege proofs.

A useful lemma on the extension of finitely indexed functions

DIRK VAN DALEN

In a constructive setting the extension problem for functions requires special attention, even in seemingly trivial cases. As a rule, some extra features have to be added in order to make the problem manageable. The following lemma is shown: every strictly monotone function on $\{a_1, \ldots, a_n\} \subseteq \mathbf{R}$ can be extended to a strictly monotonic $\overline{f}$ on $\{a_1, \ldots, a_n, a_{n+1}\}$. The problem in the case of finitely indexed sets is that the elements need not be distinct, nor is the equality decidable. Extra information is required in order to construct the image of a_{n+1}. By Brouwer's continuity theorem, every $f : \{a_1, \ldots, a_n\} \to \mathbf{R}$ is continuous, and so the given f is actually a homomorphism. By exploiting the continuity moduli (both ways) one can extend f by $f(a_{n+1})$. Fortunately one does not ned the continuity theorem (i.e. fan theorem), bacause for strongly extensional functions one can prove the continuity directly. By a spoiling argument one can prove the same extension result for $f : \{a_1, \ldots, a_{n+1}\} \to X$, where $X = \mathbf{Q}^{\#}, \mathbf{Q}^{c}$ or $\mathbf{Q}^{cc}$. ($\mathbf{Q}^{\#}$ consists of the irrationals apart from all rationals, the ***strong*** irrationals.)

The above lemma allows us to prove, using Ehrenfeucht-Fraïssé games that $\mathbf{Q}^{\#}, \mathbf{Q}^{c}$ and $\mathbf{Q}^{cc}$ are elementary equivalent substructures of $\mathbf{R}$ (w.r.t. $<$ and $\#$, the ***apartness*** relation) and that $\mathbf{R}$ is an elementary substructure of $\mathbf{R}^2$ (w.r.t. $<$).

Many-Valued Modal Logics

MELVIN FITTING

Suppose there are several experts, possibly with some dominating others. (Expert A dominates expert B if B agrees to the truth of anything A declares true.) This puts some constraints on the logic each can have. For example, if A dominates B and B believes X is false, A must also take X to be false. Consequently, the only way A can take $\neg X$ to be true is if A and every expert that A dominates takes X to be false. This gives the structure of experts the flavor of a Kripke Intuitionistic model.

Further, suppose the logical language is modal, and there is a given set of possible worlds. Each expert has his or her own opinion on the truth of atoms at worlds, and also his or her opinion on which worlds are accessible from which. All this is subject to the dominance condition mentioned above. This gives us a ***multiple expert modal model.***

Now, suppose we move to a many-valued logic by taking sets of experts, closed under dominance, as truth values. At each world of the modal model each formula is assigned one of these sets as its truth value: the set of experts who take the formula to be true at that world. In this way we get a natural notion of a many-valued modal model. The truth value of a formula at a world is taken from a many-valued logic, and also the accessibility relation is many-valued. (There have been earlier approaches to many-valued modal logic, but in all of them the accessibility relation was classical.)

The notion of a many-valued modal model can be formulated directly, in a simple way. And it can be proved that all many-valued modal models arise from multiple-expert modal models in the way described above. Further, the set of validities of a many-valued modal logic can be characterized proof-theoretically, and a completeness theorem can be proved.

Finally, each modal logic has a corresponding non-monotonic version, according to a now-standard construction. In an analogous way, the many-valued modal logic described above also has a non-monotonic version. Some standard results from the theory of non-monotonic logics extend to the many-valued version. Research on this is currently under way.

Unique Satisfiability on Horn Sets can be solved in nearly linear time

KENNETH BERMAN JOHN FRANCO JOHN SCHLIPF

The Unique Satisfiability problem is, given a Boolean expression I, does there exist a unique satisfying truth assignment for I? This problem can be solved easily in time bounded by a degree two polynomial on the length of expressions when they are known to be Horn. The idea is to use a linear time algorithm for strongly connected components on a dynamic digraph whose vertices are the atoms of I and whose edges represent implications due to the Horn clauses. A two literal clause is represented immediately in the digraph as an edge directed from its negated atom to its positive atom and other clauses (called hyperclauses) become represented by edges (we say a hyperclause is moved to the digraph) when it is found that all negated atoms in such hyperclauses are equivalent. Such an idea does not result in a linear time algorithm because of the bookkeeping necessary to determine exactly when a hyperclause should be moved to the digraph: that is, the bookkeeping necessary to determine when all the negated atoms of a hyperclause are equivalent. However, we show that hyperclauses may be moved to the

digraph early without affecting the outcome provided that the new edges are directed from the first visited negated atom of hyperclauses to the positive atoms and that the traversal of the digraph be ordered much like depth-first-search. Movement of a hyperclause to the digraph occurs when all of its negated atoms have been visited during the traversal. Since it is much easier to keep track of visiting than equivalence of the negated atoms, the result is an algorithm with complexity that differs from linear by a factor resembling $log^*(|F|)$.

A Threshold for Satisfiability

ANDREAS GÖRDT

A propositional formula is in 2–CNF (2–conjunctive normal form) iff it is the conjunction of clauses each of which has exactly two literals. We show: if $C = 1 + \epsilon$ where $\epsilon > 0$ is fixed and $q(n) >= Cn$, then almost all formulas in 2-CNF with $q(n)$ different clauses, where n is the number of variables, are unsatisfiable. If $C = 1 - \epsilon$ and $q(n) <= Cn$, then almost all formulas with $q(n)$ clauses are satisfiable. By "almost all" we mean that the probability of the set of unsatisfiable or satisfiable formulas among all formulas with $q(n)$ clauses approaches 1 as $n \to \infty$. So $C = 1$ gives us a threshold separating satisfiability and unsatisfiability of formulas in 2–CNF in a probabilistic, asymptotic sense. To prove our result, we translate the satisfiability problem of formulas in 2–CNF into a graph theoretical question. Then we apply techniques from the theory of ransdom graphs.

Computational complexity and logic of finite structures

ETIENNE GRANDJEAN

The title of this talk could have been: "Computational complexity and (descriptive) logical complexity" or "why is a class of problems an interesting class ?"

I present some personal views of this last question by giving several attempts of answer and asking other questions. I state four criteria that, in my opinion, a "good" complexity class must satisfy:

1) C must be defined by some fixed type T of machines with a fixed ressource bound; 2) C must be robust (the type T of machines can be changed but C is not changed); 3) C must be defined by a logic (with synctactic restrictions); 4) C must have natural and various complete problems.

Using these four criteria, I review six "complexity classes"; classical ones: NP, NLOGSPACE, P; personal ones: NLINEAR, DLINEAR (nondederministic and deterministic versions of linear time) and SAT–easy. In particular, all these classes, except DLINEAR and SAT–easy, can be characterized by a logic (existential second-order logic or some of its syntactical restrictions). I show that NLINEAR satisfies criteria (1–3) and a half of criterion (4) and that SAT–easy does not satisfy criteria (1–3) but fully satisfy (4). The completeness results for class NLINEAR (resp. SAT–easy) are given via reductions computable in linear time on a Turing machine (resp. Turing machine using a fixed number of free sortings).

On The Complexity of Nonmonotonic Reasoning

GEORG GOTTLOB

Complexity results for different reasoning tasks in nonmonotonic propositional reasoning are presented. In the first part of the talk, the attention is drawn to systems of nonmonotonic logic such as Reiter's default logic, Moore's autoepistemic logic, the nonmonotonic logic of McDermott and Doyle (NM1), and the nonmonotonic modal logic **N** by Marek and Truszczyński. All these logics have in common that the semantics of an initially given set of premises is explained through a corresponding set of fixed points (also called extensions or expansions). We show that for all these logics, brave reasoning is $\mathbf{\Sigma}_2^P$-complete, while cautious reasoning is $\mathbf{\Pi}_2^P$-complete. In the second part of the talk, different other formalisms of nonmonotonic reasoning, such as propositional circumscription, abduction, and various approaches to theory revision are discussed. We show that the main decision problems in these formalisms are complete for classes at the second level of the polynomial hierarchy. An exception is Dalal's approach to theory revision which we classify as $\mathbf{P}^{\mathbf{NP}[O(\log n)]}$-complete. The results of the second part of the talk are joint work with Thomas Eiter.

Evolving Algebras

YURI GUREVICH

Evolving algebras were introduced by the author a few years ago. The idea was to provide operational semantics for algorithms by strengthening Turing thesis so that algorithms can be simulated in lock-step and at the desired abstraction level. There is by now a sizable evidence showing that evolving algebras indeed provide the desired simulation. The talk is an introduction and a survey of the area. In particular, we speak about universal evolving algebras and how evolving algebras can be used for verification, correctness proofs, etc.

Stable Logic

BRIGITTE HÖSLI

Stable logic is a three-valued logic, where the third truth-value has the intention "insignificant" and where "true" and "insignificant" are the distinguished values. It is possible to characterize the tautology in stable logic as follows: A formula is a tautology iff every subformula, which arises by elimination of propositional variables, is a classical tautology. By this stability against loosing variables, the logic is called stable.

Furthermore the logic has a strong relation to the weakening rule of the sequent calculus. We obtain a calculus of classical logic only by adding the weakening rule (on the right side) to the calculus of stable logic.

The classical sequent calculus without the weakening on the left side is sound and complete w.r.t. the three-valued semantics, where "true" is the only distinguished value. The classical calculus without both weakening rules is sound and complete w.r.t. the three-valued semantics, where the truth-values are ordered.

Stable logic has many connections to other non-classical logics. So it is possible to describe the interpretation of the connectives in Lukasiewicz's logic, and we can see the truth-tables of stable logic as a counterpart to Bočvar's ones. Furthermore RM_3, the strongest logic in the family of relevance logics, has the same interpretation of the implication as stable logic.

The stable semantics is a special case of the phase semantics of Girard's linear logic. So we can derive them by a simple restriction from the interpretation of the multiplicative connectives. By the same restriction we obtain an interpretation of the exponentional connectives in stable logic. Finally, the extended stable logic is functionally complete only by adding a constant "true".

Partial fixed point theories and logic programming

GERHARD JAEGER

An adequate treatment of negation and negative information is considered as one of the most important problems in the context of logic programming. The procedural approach to negation is generally provided by the so called negation as failure rule which is fairly easy to implement but extremely delicate from the point of view of denotational meaning. Other interesting (and related concepts) are for example discussed in connection with to the closed world assumption, the completion of theories or semantics based on some kind of minimal models semantics.

Partial fixed point theories are developed in order to study the proof- theoretic aspects of (large classes) of logic programs with negation. In addition, they provide a bridge to theories of (iterated) inductive definitions and the definability theory of those. Hence the also help to exploit many results on inductive definability for logic programming and to provide a conceptually clear an perspicuous approach to several concepts in this area.

Learning Read-Once Formulas over Different Bases

MAREK KARPINSKI
(Joint work with L.Hellerstein)

We study computational complexity of learning read–once formulas over different bases.In particular we design the first polynomial time algorithm for learning read–once formulas over a threshold basis. By the result of Angluin–Hellerstein–Karpinski, 1989, on the corresponding unate class of boolean functions, this gives the first polynomial time learning algorithm for arbitrary read–once formulas over a threshold basis with negation using membership and equivalence queries. Futhermore we study the structural notion of nondegeneracy in the threshold formulas generalizing the result of Heiman–Newman–Wigderson, 1990, on the uniqueness of read–once formulas over boolean bases, and derive a negative result on learnability of nondegenerate read-once formulas over the basis (AND,XOR).

We discuss also the problem of learning depth-k decision trees, and give a polynomial time algorithm for this problem.

An algebraic framework for logic programming and the wellfounded model

GERARD R. RENARDEL DE LAVALETTE
(joint work with Catholijn Jonker)

This talk is about propositional general logic programs (negated atoms allowed in the bodies of clauses). An interesting question is whether the meaning of such a program can be captured by a single model, thereby incorporating some kind of negation as failure. Several such models have been proposed: the perfect model, the stable model (both with two truth values), the partial stable model and the wellfounded model (both with the additional truth value ***undefined***). Of these only the wellfounded model (introduced by Van Gelder, Ross and Schlipf in 1988) is at the same time universal, unique and tractable.

We present simple algebraic definitions of general logic programs and related notions, abstracting from the usual representation by clauses. This framework allows for a compact definition of the wellfounded model. We also present an alternative definition using two fixpoint constructions, which leads to a quadratic algorithm (using the linear-time algorithm for the minimal model of a collection of Horn clauses by Dowling & Gallier and Minoux). This algorithm has essentially been developed by Witteveen.

ILFA: A project in experimental logic computation

ANDREAS FLÖGEL HANS KLEINE BÜNING JÜRGEN LEHMANN THEODOR LETTMANN

ILFA is a library of general reusable software components for logic processing. ILFA stands for **I**ntegrated **L**ogical **F**unctions for Advanced **A**pplications and resulted from a research project between IBM and the university of Duisburg.

The realisation in C allows the efficient use of logical methods in many different computer environments. ILFA contains a multitude of logic algorithms as well as a class–based realisation for important sublanguages of logic. Special attention has been put on an intuitive user interface for use and combination of the library components.

It has been shown that logic processing, e.g. the manipulation of logic expressions, can practically be used as a tool in computer science. Also for the study of the theoretical behaviour of logical algorithms, the experimentation with different variations has proved to be useful. The most practical work of implementations of logic programming has been done to build special efficient theorem provers. (e.g if we consider Prolog as a kind of theorem prover). But, in general, there exists a wealth of (theoretical) knowledge about logic algorithms. Many of these logic algorithms have been implemented, but it

is difficult to build upon the implementationbecause they are stand–alone and usually not specially designed for reuse.

ILFA is a step towards a software library of efficient implementations for a wide range of logic algorithms. The top level structure of the library can be pictured by several classes (or areas).The classes stand for important sublanguages of logic processing and the belonging logic algorithms for the classes. These areas are loosely coupled by a star–architecture in order to allow the transformation from one representation to another.

In the ILFA–system, we have made special efforts in easing the use of the ILFA–components. The user interface should support the different goals of ILFA. The first goal is the fast experimentation and work with the algorithms of the system. Without special knowledge of a programming language, the user can easily start to perform tasks in this area. The second kind of interface is the toolbox environment. Here, different components can be combined in a more flexible way. This will mainly be done by means of the black box methapher, i.e. looking upon algorithms and objects stored in data structures of the library as building blocks of new algorithms. The environment gives an iconic interface of the building blocks. It also supports the programming of meta algorithms. For example, an application can generate n different provers which work with variations on a knowledge base. The application can collect and interpret the results of the different provers, write them into new knowledge bases and redistribute the knowledge bases on possibly modified new provers.

The enhancement of the system with further areas is an ongoing work.

On the average case complexity of algorithms in general and on SAT in particular

JOHANN A. MAKOWSKY
(in collaboration with Avraham Sharell)

We propose a suitable definition for the *average lower bounds* and show that it complements exactly the existing of the corresponding definitions for average upper bounds. To discuss the notion of probabilistic lower bounds in more depth, we propose several properties which allow us to evaluate candidate definitions. These properties are: *Transitivity properties, honesty with respect to worst case bounds, consistency with probabilistic upper bounds, dichotomy with respect to bounding functions and robustness under small changes of the underlying probability function.*

To illustrate our machinery we discuss most existing results on the probabilistic analysis of resolution based algorithms for SAT and derive new statements on the average behaviour of resolution.

References:

J.A. Makowsky and A. Sharell, On the average case complexity of SAT for symmetric distributions, Technical report TR 739, Department of Computer Science, Technion–Israel Institute of Technology, 1992

A. Sharell and J.A. Makowsky, Probabilistic lower bounds for average case complexity, Technical report TR 746, Department of Computer Science, Technion–Israel Institute of Technology, 1992

A modality-free interpretation of classical logic into linear logic

SIMONE MARTINI ANDREA MASINI

Several translations of intuitionistic logic into Girard's linear logic have been proposed. Almost all of them agree on a heavy use of *modalities* (! and ?) in order to mimick weakening and contraction rules, the notable exception being the one give by Lincoln *et al.* in LICS91, where a translation of the intuitionistic implicational logic into the intuitionistic fragment of multiplicative-additive linear logic (IMALL) is given that does not use modalities. We show how the full propositional classical logic can be translated into the Multiplicative-Additive fragment of propositional linear logic *without modalities* (MALL), and without factorizing the interpretation through a translation of classical logic into intuitionistic logic. The translation naturally defines a fragment of MALL for which the decision problem is coNP-complete (for full MALL is PSPACE-complete, while for the full logic with modalities is undecidable). The translation is an "asymmetrical interpretation" and, as such, it does not validate the cut-rule (that is, it translates cut-free proofs in a suitable formal system for classical logic into cut-free linear proofs).

Analyzing and Manipulating Boolean Functions by Means of Restricted Branching Programs

CHRISTOPH MEINEL

We investigate the question whether and to what extend the solution of central tasks of digital logic circuit design of a given Boolean function f benefits from a representation of f in terms of certain restricted branching programs.

In detail we investigate

- satisfiability and tautology,
- equivalence,
- simultaneous satisfiability, and
- binary synthesis.

The work was done jointly with Jordan Gergov.

Algebraic Semantics of Rewriting Terms and Types

KARL MEINKE

We present a universal algebraic framework for rewriting terms and types over an arbitrary equational specification of types and typed combinators. Equational type specifications and their initial algebra semantics were introduced in Meinke [1991]. For an arbitrary equational type specification (ε, E) we prove that the corresponding rewriting relation $\xrightarrow{R(\varepsilon,E)^*}$ coincides with the provability relation $(\varepsilon, E) \vdash$ for the equational calculus of terms and types. Using completeness results for this calculus we deduce that rewriting for ground terms and ground types coincides with calculation in the initial model $I(\varepsilon, E)$ of the type specification.

References

K. Meinke, *Equational specification of abstract types and combinators*, to appear in G. Jaeger (ed), Proc. Computer Science Logic '91, Lecture Notes in Computer Science, Springer Verlag, Berlin, 1991.

Normal forms in many-valued logic

DANIELE MUNDICI

Many-valued automated deduction is a rapidly expanding field of research, and deep connections are being found between error correcting codes, AF C^*-algebras, and the infinite-valued sentential calculus of Łukasiewicz. Surprisingly enough, little attention has been devoted to normal forms. We present an effective normal form reduction technique: given any proposition p in the infinite-valued calculus, p is decomposed into a disjunction of basic constituents; the latter have a simple geometric form, and are in fact the Schauder hats of the triangulation induced by p on its domain. Since logical equivalence in the infinite–valued calculus is stronger than all its n–valued counterparts, our reductions are automatically valid for all n–valued calculi. As a by–product, we obtain the first constructive proof of the fundamental McNaughton theorem (JSL volume of 1951) stating that, up to logical equivalence, propositions in the infinite-valued calculus coincide with continuous piecewise linear functions all of whose pieces have integral coefficients. Our constructions are elementary; the key tool is Minkowski's convex body theorem.

Undecidability of the Horn clause Implication Problem

JERZY MARCINKOWSKI LESZEK PACHOLSKI

We prove that the problem "given two Horn clauses $\mathcal{H}_1 = (\alpha_1 \wedge \alpha_2 \rightarrow \beta)$ and $\mathcal{H}_2 = (\gamma_1 \wedge \ldots \wedge \gamma_k \rightarrow \delta)$, where $\alpha_1, \beta, \gamma_i, \delta$ are atomic formulas, decide if $\mathcal{H}_2$ is a consequence of $\mathcal{H}_1$" is not recursive.

The theorem follows from the series of more or less technical lemmas.

Definition 1. For a Horn clause $\mathcal{H} = (\alpha_1 \wedge \alpha_2 \rightarrow \beta)$, and a set G of ground clauses a G-$\mathcal{H}$-derivation tree is a tree labelled by unit clausses in such a way, that for each node t there exists a substitution σ with the property that the left and the right son of the node t are labelled by $\sigma(\alpha_1)$ and $\sigma(\alpha_2)$ respectively, and t is labelled by $\sigma(\beta)$, and moreover the leaves are labelled with elements of G.

Lemma 2. There exists a Horn clause $\mathcal{H} = (\alpha_1 \wedge \alpha_2 \rightarrow \beta)$, and a finite set G of ground unit clauses such that it is undecidable if for a given word w, there exists a finite G-$\mathcal{H}$-derivation tree with a branch w.

The next two lemmas have a technical character and say, that it is possible to force a derivation tree to contain a given branch (**Forcing Lemma**) and to hide the large uncontrolled term that appears in the root of a derivation (**Hiding Lemma**).

Reasoning about Knowledge

ROHIT PARIKH

This has become a rich area in the last few years. We give some examples.

1. Dialogues: We show how people learn from dialogues by modelling such dialogues in Kripke structures. There is a resemblance to the Cantor-Bendixson theorem which can be made precise. We show that some facts can be learned in certain dialogues only at a transfinite stage.

However, if we change the rules to: win one dollar for a correct guess, lose a thousand dollars for an incorrect guess. Then even though knowledge itself may take a transfinite amount of time, a profitable guess can always be made after a finite amount of time.

2. Non-monotonicity: If I tell you that $a \times b = 12$, where a, b are understood to be integers with $1 < a \leq b$, then you do not know what a is, since a may be 2 or 3. However, if I tell you also that a is even, then you know that a must be 2. Thus if $\Gamma = \{K_y(a \times b = 12), K_y(1 < a \leq b, a, b \in N\}$ then $\Gamma \vdash \neg K_y(a = 2)$ but $\Gamma, K_y(a \text{ is even}) \vdash K_y(a = 2)$ and hence we cannot have $\Gamma, K_y(a \text{ is even}) \vdash \neg K_y(a = 2)$.

Thus we have non-monotonicity as soon as knowledge enters explicitly. McCarthy has proposed the rule, "if you cannot prove $K_y(A)$ from Γ, then deduce $\neg K_y(A)$ from Γ". We develop a model theory (using largest models) and show a completeness result.

3. Speaker dependence of Language: A slide containing colored squares was projected on a screen. The audience was asked to write down how many red squares and how many blue squares they saw. There was wide variation. In particular, the number of blue squares seen varied from 0 to 27. Thus everyone interprets "blue" in his own way, though of course there is a rough correspondence between different usages.

If Alice tells Bob, "my book is blue", she means that her book is blue_A but he will take her to mean that her book is blue_B. We show how this information is helpful to him in spite of this in many cases.

4. Reducing Topology to the Logic of Knowledge (joint work with L. Moss and K. Georgatos): We show how topological notions can be expressed in a language with two modalities: K for knowledge and $\Box$ for effort. We get a very nice logic which is currently under investigation.

A Formal Model for Gödel

ELVINIA RICCOBENE

(joint work in progress with Egon Börger)

Gödel is a logic programming language which is intended to be a declarative successor to Prolog. It has functionality and expressiveness similar to Prolog, but greatly improved declarative semantics compared with it; in fact Gödel does not have the "impure" non-logical aspects of Prolog, like the absence of occur-check, the use of unsafe negation, the use of non-logical predicates (*var, nonvar, assert, retract*), the cut operator, ecc.

Gödel is a strongly typed language based on many-sorted logic with parametric polymorphism. It has a module system and a flexible computational rule by a **DELAY** operator. The Prolog *cut* operator is replaced by a **commit** operator, which generalises the *bar commit* of concurrent logic programming languages. Gödel has also very powerful meta-logical facilities.

We propose a mathematically precise but natural formalisation of Gödel. The abstract (operational) model is based upon Gurevich's notion of *evolving algebras*.

Since the Gödel computational rule is a generalised form of SLDNF resolution, we basically base our model on the *Prolog Tree Algebra* defined by Börger & Rosenzweig as formal model for the Prolog language.

We mantain all the *Prolog Tree Algebra* transition rules (for used defined predicates) appropriately changed in order to formalise the non-sequentiality of Gödel computational rule both in the *selection phase* (selecting the literal to compute into a conjunction of goals) and in the *search phase* (selecting a possible candidate clause to reduce the current call).

We obtain a simple description of the procedural semantics of the **commit** operator and the pruning phase, and we also give a complete formalisation of the negative (ground) literals and the *conditionals* computation.

Our future goal is to refine our description to a description of the extension of the Gödel model with the formalisation of meta-programming facilities of the language. We also plan to look for a parallel version of the model, using our previous work done for an *evolving algebra* specification of PARLOG.

Evolving algebras and process calculi

DEAN ROSENZWEIG

(Joint work with Paola Glavan)

The notion of 'concurrent evolving structure' of Gurevich & Moss is made more explicit, and linked to usual algebraic calculi of processes.

Process calculi may be viewed as *descriptions of behaviour*, while evolving algebras can be understood as (very) *abstract machines*. It is then natural that we can describe behaviour of machines, and build machines which exhibit given behaviour.

In particular, we show how to simulate concurrent evolving structures by processes of value–passing CCS, and vice versa. The simulations are correct in the following sense:

If, composing the simulations, *process* $\longrightarrow$ *algebra* $\longrightarrow$ *process*, we get from P to Q, they will be linked by the following relation:

$P \mathcal{R} Q$ if $\forall \alpha \in \mathcal{L}(P)\ \exists rep(\alpha) \subset \mathcal{L}(Q)^*$ such that

$$P \xrightarrow{\alpha} P' \quad \text{iff} \quad Q \stackrel{t}{\Longrightarrow} Q'$$

for some $t \in rep(\alpha)$, and $P' \mathcal{R} Q'$.

Since processes and evolving algebras communicate by different primitive actions (handshake *cum* value–passing vs. updating 'blackboard–like' shared structures), we loose in abstraction level at both simulation steps, arriving thus at the problem of comparing processes at different levels of abstraction.

Functional Completeness in Temporal Logic

G. Ligozat M. de Rougemont

For a model of time, a set of operators is *functionally complete* if, for any first-order formula with k free variables, there exists an equivalent temporal formula with k reference points.

A theory has a finite H-dimension p, if any first-order formula is equivalent to a formula with at most p bound variables. Gabbay (1979) showed that there exists a set of complete operators iff the theory axiomatizing the models of time has a finite H-dimension.

Immerman & Kozen (1989) gave a game interpretation of the H-dimension as follows: generalize the ki–Fraisse games between two structures U, V and two players I and II, to n rounds, starting from k-vectors $u \in U$ and $v \in V$. Let $G(u, v, k, n)$ if II wins n rounds. The theory has H-dimension k, if for all structures U, V, starting vectors u, v and for all n, $G(u, v, k, n)$ implies $G(u, v, k', n)$ with $k' > k$.

We present the classical applications of this result to linear and branching time, and generalize it to other models of time, multi-trees, and some lattices.

Proving total correctness of programs in weak second-order logic

Rudolf Berghammer Birgit Elbl Ulf Schmerl

A purely syntactical but nevertheless handy definition of the predicate transformer *wp* is presented. Weak second-order logic is shown to be sufficient to formalize the weakest precondition for an imperative programming language similar to Dijkstra's language of guarded commands. It is demonstrated how to express and prove important properties in this logic.

Nonmonotonic reasoning

PETER SCHMITT

In this talk we present a survey of the approach to nonmonotonic reasoning developed by S. Kraus, D. Lehmann and M. Magidor. We concentrate on the notions of preferential and rational closure of a given knowledge base, review the motivations that lead to the axiomatic approach, discuss the proposed axioms (rules), introduce a model theoretic semantics and quote the known completeness theorems. We conclude with the following list of future research problems: 1. Extend the notions of preferential and rational closure to full first order logic. 2. Find a model theory for the rational closure. 3. Find a sufficiently concrete area of applicability for nonmonotonic logic.

Machine Checked Normalization Proofs for Typed Combinator Calculi

JAN SMITH

We present formalized normalization proofs of two calculi: simply typed combinators and a combinator formulation of Gödel's system T. The proofs are based on Tait's computability method and formalized in Martin-Löf's set theory. The motivation for doing these formalizations is to obtain machine checked normalization proofs. The proofs presented in this talk have been checked using the the logical framework implementation Alf, developed at Göteborg.

The first theory we discuss is the simply typed combinators. It is chosen because of its simplicity: we want a machine checked proof which avoids as much as possible syntactical problems and concentrates on the computability method.

Tait treated a combinator formulation of Gödel's T. Although that is a much more powerful theory than the simply typed combinators, the normalization proof for Gödel's T is, from the formal point of view, a straightforward extension of that for the simply typed combinators.

This talked is based on a joint paper with Veronica Gaspes.

On the Complexity of Some Decision Problems in Programming

DIETER SPREEN

One of the central problems in programming is the correctness problems, *i.e.* the question whether a program P computes a given function. If the function is given by a program Q which is already known to be correct, then the problem can be reduced to the question whether the two programs P and Q are equivalent. Sometimes, one only wants to know the correctness of the program P with respect to a given set of input data. If Q is correct with respect to these data, then it is sufficient to decide whether the function computed by P extends the function computed by Q.

In order to deal with these problems mathematically, one has to fix a semantical framework. We use Scott domains. Let $(D, \sqsubseteq)$ be a Scott domain and x a canonical indexing of its computable elements, then we consider the decision problems "$x_i = d$?", for some given $d \in D$, "$x_i = x_j$?", and "$x_i \sqsubseteq x_j$?". As it follows from the generalized Rice/Shapiro theorem and a characterization of the index sets of non-Scott-open sets these problems are undecidable. It is the aim of this work to classify their complexity. As it is shown, the question whether $x_i = d$ is Π_2^0-complete, if d is nonfinite. In the other case the problem is recursively isomorphic to the halting set K, if d is maximal with respect to the domain order. It is recursively isomorphic to the complement $\overline{K}$ of the halting set, if $d = \perp$, and it is recursively isomorphic to $K \times \overline{K}$ in the remaining case. As it is well known, K is Σ_1^0-complete, $\overline{K}$ is Π_1^0-complete, and $K \times \overline{K}$ is Σ_2^{-1}-complete where $\{ \Sigma_n^{-1}, \Pi_n^{-1} \mid n \geq 1 \}$ is the Boolean hierarchy. This completely classifies the difficulty of deciding the question "$x_i = d$?".

For the remaining problems "$x_i = x_j$?" and "$x_i \sqsubseteq x_j$?" it is shown that they are recursively isomorphic. Moreover, they are Π_2^0-complete, if the domain contains a computable nonfinite element.

If all computable elements in D are finite, then all ascending chains of finite elements in D have to be finite. In the case that the lengths of all such chains are bounded and m is the maximal length, both "$x_i = x_j$?" and "$x_i \sqsubseteq x_j$?" are $\Pi_{2(m-1)}^{-1}$-complete. In the other case, it is shown that both problems are in $\Sigma_2^0 \cap \Pi_2^0 \setminus \bigcup \{ \Pi_n^{-1} \mid n \geq 1 \}$. Moreover, two problems S_1 and S_2 in this class are given such that $S_1 \leq_1$ "$x_i \sqsubseteq x_j$?" $\leq_1 S_2$.

Cut-elimination and negation as failure

ROBERT F. STÄRK

What is the semantics of 'Negation-as-Failure' in logic-programming? We try to answer this question by proof-theoretic methods. We have developed a rule based sequent calculus for negation as failure. Given any program P, a sequent Γ is provable in the calculus if, and only if, it is true in all three-valued models of the completion of P. The calculus is exactly the sequent calculus for the classical completion of a program but without axioms of the form $\Gamma, A, \neg A$. The reason that we do not use axioms of the form $\Gamma, A, \neg A$ is that they imply that a formula A has to be true or false; in terms of logic programming this means that an atom A must succeed or fail, which in general is not true. It is easy to transform SLDNF-computations into sequent proofs. For certain classes of programs the converse is also possible. We give a sufficient and necessary condition on a program such that this is possible. Via cut-elmination the complexity of a sequent proof is bounded to sequents constructed from equations and literals only. Such proofs can then be converted into SLDNF-computations. We obtain the main theorem that a normal program is ***negation-complete*** if, and only if, it has the ***cut-property***. From this theorem we can derive a very strong completeness result for SLDNF-resolution. Finally we define a decidable class of logic programs which have the cut-property. This class contains the definite programs and the allowed programs.

Fixpoint Logics, Relational Machines and Computational Complexity

VICTOR VIANU
(joint work with S. Abiteboul and M.Y. Vardi)

We establish a general connection between fixpoint logic and complexity. On one side, we have fixpoint logic, parameterized by the choices of 1st–order operators (inflationary or noninflationary) and iteration constructs (deterministic, nondeterministic, and alternating). On the other side, we have the complexity classes between P and EXPTIME. Our parameterized fixpoint logics capture the complexity classes P, NP, PSPACE, and EXPTIME, but equality is achieved only over ordered structures.

There is, however, an inherent mismatch between complexity and logic—while devices work on encodings of structures, logic is applied directly to the underlying structures. To overcome this mismatch, we develop a theory of relational complexity, which bridges the gap between standard complexity and fixpoint logic. On one hand, we show that questions about containments among standard complexity classes can be translated to questions about containment among relational classes. On the other hand, the expressive power of fixpoint logic can be precisely characterized in terms of relational complexity classes. This tight three–way relationship among fixpoint logics, relational complexity, and standard complexity yields in a uniform way logical analogs to all containments among the complexity classes P, NP, PSPACE, and EXPTIME. The logical formulation shows that some of the most tantalizing questions in complexity theory boil down to a single question: the relative power of inflationary versus noninflationary 1st-order operators.

This paper appeared in Proc. Conf. on Structure in Complexity Theory, 1992. It is a followup of the paper "Generic Computation and Its Complexity" (S.Abiteboul and V.Vianu), which appeared in STOC 91.

Abstract Data Types and Locally Reflective Subcategories

J.ADAMEK H.VOLGER

We have obtained a localized version of the abstract initial semantics which is used in the theory of abstract data types. One considers the following situation: a category K (the category of 'structures') and a full, isomorphismclosed subcategory A (the category of 'models') and one is looking for characterization results of the following type.

Theorem. If the ambient category K has Δ-limits then:
A is Δ-*reflective* in K i.e. for each $X \in \mathcal{K}$ the category x\A admits enough Δ-*initial objects* iff A is closed under Δ-*limits* and A satisfies the Δ-solution set condition.

The well known global version uses *arbitrary limits* (i.e. equalizers and products) and *initial* objects in $X\backslash\mathcal{A}$. The conditional version of Y.Diers uses *connected limits* (i.e. equalizers and pullbacks) and *conditional initial* objects i.e. objects initial in their connected component of $X\backslash\mathcal{A}$. Our localized version uses *simply connected limits* (i.e. pullbacks) and *locally initial* objects i.e. objects remaining

initial in all their localizations. It can be shown that A is locally reflective in K iff all the localizations $\mathcal{A}/A$ are reflective in $\mathcal{K}/A$ iff $\mathcal{A}$ is conditionally pseudoreflective in K. The qualifier 'pseudo' means replacing uniqueness by uniqueness up to an isomorphism. – It should be noted that categories with pullbacks have been used in the theory of domains under the name of stable domains.

In the case $\mathcal{A} = Mod(T)$ with T a firstorder theory it suffices to consider finite limits and the solution set condition can be removed. A syntactical characterization in the global case is due to Volger('79) and in the conditional case was given by M.Hébert ('89). For the localized version Hébert recently announced a syntactical characterization result. – In the firstorder case there is another characterization result concerning equalizers. However, we were not able to prove this result in the general situation.

Object modelling using λ-calculus for information systems

JIŘÍ ZLATUŠKA

For information system modelling, the relational model provided a framework based on logic, essentially Datalog, providing for declarative data description and manipulation. Presently, object-oriented models loose the implementation-independent background, being predominantly based on programming constructs.

We argue for a model based on a variant of a simply typed λ-calculus, enriched by product and sum types, and corresponding reductions. Using functions as the basic construct, the rule-manipulating nature of λ-calculus conversions can account for uniform expression of several features of object-oriented environment.

Special classes of expressions can be chosen for encoding relational structures (implementational data types) and functional structures (object-based types), with correspondence between them given by mapping pairs carrying out transformation of one type of description into another. The basic idea is to use the concept of schema transformation for representing various classes of data manipulation: retrieval, structure mapping, updates, and transactions. Operations modifying the state of the system result in terms defining schema mapping, having a counterpart in mappings between functional environments, representing these state-changing operations as transformations of the underlying interpretation. Multi-layered structure is obtained by composing schema transformations, adjusting to the level of abstraction needed for particular purpose. As a result, semi first-order logic-based language for object manipulation is obtained with encapsulated data operations provided by combinators, providing tools for veryfying implementation of the operations wrt declarative/object-oriented specification, and a tool for sufficiently abstract representation of objects, object identity, and data abstractions independently of the implementation structures.

Dagstuhl-Seminar 9229:

List of Participants

Miklos **Ajtai**
IBM Almaden Research Center
650 Harry Road
San Jose CA 95120-6099, USA
ajtai@almaden.ibm.com
tel.: +1-408-927-18 52

Klaus **Ambos-Spies**
Universität Heidelberg
Mathematisches Institut
Im Neuenheimer Feld 288
W-6900 Heidelberg 1, Germany
G77@DHDURZ1.BITNET
tel.: +49-6221-562673

Egon **Börger**
Università di Pisa
Dipartimento di Informatica
Corso Italia 40
I-56125 Pisa, Italy
boerger@di.unipi.it
tel.: +39-50-51 02 30

Corrado **Boehm**
Universita di Scienze Dell'Informazione
Via Salaria 113
Roma, Italy
boehm@vaxrma.DNET.NASA.GOV

Sam **Buss**
University of California at San Diego
Department of Mathematics
La Jolla CA 92093-0112, USA
sbuss@sbuss.ucsd.edu
tel: 619-534-6455

Peter **Clote**
Boston College
Dept. of Computer Science
140 /Commonwealth Ave.
Chestnut Hill MA 02167, USA
clote@bcuxsz.bc.edu
tel.: +1-617-522-43 33

Dirk **van Dalen**
Rijksuniversiteit Utrecht
Department of Philosophy
Heidelberglaan 8
NL-3584 CS Utrecht
The Netherlands
Dirk.vanDalen@phil.ruu.nl
tel.: +31-30-53 18 34

Melvin **Fitting**
City Univ. of N. Y.
Dept. of Math. and Comp. Science
Bedford Park Blvd. West
Bronx NY 10468, USA
MLFLC@CUNYUM.CUNY.EDU
tel: 212 960 8117

John **Franco**
University of Cincinnati
Department of Computer Science
Mail Location 0008
822 Old Chemistry Bldg.
Cincinnati OH 45221 - 0008, USA
franco@ucunix.san.uc.edu
tel.: +1-513-556-18 17

Andreas **Goerdt**
Universität GH Paderborn
FB 17 - Mathematik/Informatik
Warburgerstr. 100
W-4790 Paderborn, Germany
goerdt@uni-paderborn.de
tel.: +49-5251-60-33 51

Georg **Gottlob**
Technische Universität Wien
Inst. für Informationssysteme
Paniglgasse 16
A-1040 Wien, Austria
gottlob@vexpert.dbai.tuwien.ac.at
tel.: +43-1-58801-61 20

Etienne **Grandjean**
Université de Caen
Laboratoire d'Informatique
U.E.R. de Sciences
F-14 032 Caen, France
grandjean@univ-caen.fr
tel.: +33-31.45.59.67

Yuri **Gurevich**
The University of Michigan
EECS Department
Ann Arbor MI 48109, USA
gurevich@zip.eecs.umich.ed

Brigitte **Hoesli**
ETH - Zürich
Institut für Theoretische Informatik
ETH-Zentrum
CH-8092 Zürich, Switzerland
hoesli@inf.ethz.ch
tel.: +41-1-2 54 73 77

Gerhard **Jäger**
Universität Bern
Inst. für Informatik und
Angewandte Mathematik
Länggassstr. 51
CH-3012 Bern, Switzerland
jaeger@at.iam.unibe.ch
tel.: +41-31-65 85 60

Marek **Karpinski**
Universität Bonn
Institut für Informatik
Römerstr. 164
W-5300 Bonn 1, Germany
karpinski@cs.uni-bonn.de
tel.: +49-228-550-224

Hans **Kleine Büning**
Universität GH Paderborn
FB 17 - Mathematik/Informatik
Warburgerstr. 100
W-4790 Paderborn, Germany
kbcsl @uni-paderborn.de
tel.: +49-5251-60 33 61

Fred **Kröger**
Universität München
Institut für Informatik
Theresienstr.39
W-8000 München 2, Germany
kroeger@.informatik.uni-muenchen.de
tel.: +49-89-23 94 44 64

Theo **Lettmann**
Universität GH Paderborn
FB 17 - Mathematik/Informatik
Warburgerstr. 100
W-4790 Paderborn, Germany
lettmann@uni-paderborn.de
tel.: +49-5251-60-33 39

Janos **Makowsky**
Technion
Computer Science Departement
Technion City
Haifa 32000, Israel
janes@cs.technion.ac.il
tel.: +972-4-29 43 58

Simone **Martini**
Università di Pisa
Dipartimento di Informatica
Corso Italia 40
I-56125 Pisa, Italy
martini@di.unipi.it
tel: +39-50-510 255

Christoph **Meinel**
Universität Trier
Fachbereich IV- Informatik
Postfach 3825
W-5500 Trier, Germany
meinel@uni-trier.de
tel.: +49-651-201-2827

Karl **Meinke**
University of Swansea
Computer Science Division
Singleton Park
Swansea SA2 8PP, Great Britain
cskarl@uk.ac.swan.pyr
tel.: +44-792-29 56 63

Daniele **Mundici**
Universitá degli Studi di Milano
Dip. Scienze dell' Informazione
via Comelico 39-41
I-20135 Milano, Italy
mundici@imiucca.csi.unimi.it
tel.: +39-2-55 00 62 47

Leszek **Pacholski**
Institute of Mathematics
Kopernika 18
PL-51 617 Wroclaw, Polonia
pacholsk@plwruw11.bitnet
tel: +48-71-25 1271

Rohit **Parikh**
CS CUNY, Graduate Center
33 W 42nd Street
New York NY 10036, USA
RIPBC@CUNYVM.CUNY.EDU
tel: 212-642-2249

David A. **Plaisted**
University of North Carolina
Department of Computer Science
352 Sitterson Hall
Chapel Hill NC 27599 - 3175, USA
plaisted@cs.unc.edu
tel: 919-962-1751

Gerard R. **Renardel de Lavalette**
University of Groningen
Department of Computning Science
Blauwborgje 3
NL-9700 AV Groningen, The Netherlands
grl@cs.rug.nl
tel.: +31-50 63 71 28/63 39 39

Elvinia **Riccobene**
Università di Catania
Dipartimento di Matematica
Viale Andrea Doria 6
I-95125 Catania,Italy
riccobene@mathct.cineca.it
tel: +39-95-330533

M.M. **Richter**
Universität Kaiserslautern
FB Informatik
Postfach 3049
W-6750 Kaiserslautern, Germany

Mario **Rodriguez Artalejo**
Universidad Complutense de Madrid
Departamento de Informática y
Automática
E-28040 Madrid, Spain
mario@emducm11.bitnet
tel.: +34-1-3 94 45 26

Dean **Rosenzweig**
Sveucilista of Zagreb
Fakultet Strojarstva I Brodogradnje
Dure Salaja 5
41 000 Zagreb, Croatia
dean.rosenzweig@uni-zg.ac.noprmd.mail.yu
tel: +38-41-611 944

Michel **de Rougemont**
Université Paris Sud
Laboratoire de Recherche en Informatique
Bât 490
F-91405 Orsay Cedex, France
mdr@lri.fr
tel.: +33-69.41.64.63

Ulf R. **Schmerl**
Universität der Bundeswehr München
Fakultät für Informatik
Werner-Heisenberg-Weg 39
W-8014 Neubiberg, Germany
schmerl@informatik.unibw-muenchen.de
tel: 089-6004-2243

Peter M. **Schmitt**
Universität Karlsruhe
Fakultät für Informatik
Am Fasanengarten
W-7500 Karlsruhe, Germany
pschmitt@ira.uka.de
tel.: +49-721-6 08 40 00

Jan **Smith**
Chalmers University of Technology
Department of Computer Sciences
S-41296 Göteborg, Sweden
smith@cs.chalmers.se

Ewald **Speckenmeyer**
Heinrich-Heine Universität
Fachbereich Mathematik / Informatik
Universitätsstraße 1
D-4000 Düsseldorf 1, Germany
speckenm@cs.uni-duesseldorf.de
tel.: +49-211-3 11 37 10

Dieter **Spreen**
Universität - GH - Siegen
Theoretische Informatik
Postfach 10 12 40
5900 Siegen, Germany
spreen@hrz.uni-siegen.dbp.de
tel.: +49-271-740-23 00/3165

Robert F. **Stärk**
Universität Bern
Inst. für Informatik und
Angewandte Mathematik
Länggassstr. 51
CH-3012 Bern, Switzerland
staerk@iam.unibe.ch
tel.: +41-31-65 85 58

Victor **Vianu**
University of California at San Diego
Dept. of Computer Science
CSE C-0114
La Jolla CA 92093-0114, USA
vianu@cs.ucsd.edu
tel.: +1-619-534-62 27

Hugo **Volger**
Universität Passau
Fachbereich Mathematik / Informatik
Innstraße 33
W-8390 Passau, Germany
volger@fmi.uni-passau.de
tel.: +49-851-50 93 49

Jiri **Zlatuska**
Masaryk University
Institute of Computer Science
Buresova 20
CS-601 77 Brno, Czechoslovakia
zlatuska@cspuni12.bitnet
tel: +42-5-753456

Zuletzt erschienene und geplante Titel:

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 17, 1.-5.7.1991 (9127)

A. Karshmer, J. Nehmer (editors):
Operating Systems of the 90s and Beyond, Dagstuhl-Seminar-Report; 18, 8.-12.7.1991 (9128)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization, Dagstuhl-Seminar-Report; 19, 26.8.-30.8.91 (9135)

T. Lengauer, R. Möhring, B. Preas (editors):
Theory and Practice of Physical Design of VLSI Systems, Dagstuhl-Seminar-Report; 20, 2.9.-6.9.91 (9136)

F. Bancilhon, P. Lockemann, D. Tsichritzis (editors):
Directions of Future Database Research, Dagstuhl-Seminar-Report; 21, 9.9.-12.9.91 (9137)

H. Alt , B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 22, 07.10.-11.10.91 (9141)

F.J. Brandenburg , J. Berstel, D. Wotschke (editors):
Trends and Applications in Formal Language Theory, Dagstuhl-Seminar-Report; 23, 14.10.-18.10.91 (9142)

H. Comon , H. Ganzinger, C. Kirchner, H. Kirchner, J.-L. Lassez , G. Smolka (editors):
Theorem Proving and Logic Programming with Constraints, Dagstuhl-Seminar-Report; 24, 21.10.-25.10.91 (9143)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures, Dagstuhl-Seminar-Report; 25, 4.11.-8.11.91 (9145)

A. Dress, M. Karpinski, M. Singer(editors):
Efficient Interpolation Algorithms, Dagstuhl-Seminar-Report; 26, 2.-6.12.91 (9149)

B. Buchberger, J. Davenport, F. Schwarz (editors):
Algorithms of Computeralgebra, Dagstuhl-Seminar-Report; 27, 16.-20.12.91 (9151)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13..-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.02.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Modelling with Programmed Machines, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13.-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.2.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Information-technological Models, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

N. Habermann, W.F. Tichy (editors):
Future Directions in Software Engineering, Dagstuhl-Seminar-Report; 32; 17.2.-21.2.92 (9208)

R. Cole, E.W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms; Dagstuhl-Seminar-Report; 33; 2.3.-6.3.92 (9210)

P. Klint, T. Reps, G. Snelting (editors):
Programming Environments; Dagstuhl-Seminar-Report; 34; 9.3.-13.3.92 (9211)

H.-D. Ehrich, J.A. Goguen, A. Sernadas (editors):
Foundations of Information Systems Specification and Design; Dagstuhl-Seminar-Report; 35; 16.3.-19.3.9 (9212)

W. Damm, Ch. Hankin, J. Hughes (editors):
Functional Languages:
Compiler Technology and Parallelism; Dagstuhl-Seminar-Report; 36; 23.3.-27.3.92 (9213)

Th. Beth, W. Diffie, G.J. Simmons (editors):
System Security; Dagstuhl-Seminar-Report; 37; 30.3.-3.4.92 (9214)

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 5.4.-9.4.92 (9215)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 39; 22.06.-26.06.92 (9226)

E. Börger, Y. Gurevich, H. Kleine-Büning, M.M. Richter (editors):
Computer Science Logic, Dagstuhl-Seminar-Report; 40; 13.07.-17.07.92 (9229)

J. von zur Gathen, M. Karpinski, D. Kozen (editors):
Algebraic Complexity and Parallelism, Dagstuhl-Seminar-Report; 41; 20.07.-24.07.92 (9230)

F. Baader, J. Siekmann, W. Snyder (editors):
6th International Workshop on Unification, Dagstuhl-Seminar-Report; 42; 29.07.-31.07.92 (9231)

J.W. Davenport, F. Krückeberg, R.E. Moore, S. Rump (editors):
Symbolic, algebraic and validated numerical Computation, Dagstuhl-Seminar-Report; 43; 03.08.-07.08.92 (9232)

R. Cohen, W. Wahlster (editors):
3rd International Workshop on User Modeling, Dagstuhl-Seminar-Report; 44; 10.-14.8.92 (9233)

R. Reischuk, D. Uhlig (editors):
Complexity and Realization of Boolean Functions, Dagstuhl-Seminar-Report; 45; 24.08.-28.08.92 (9235)

Th. Lengauer, D. Schomburg, M.S. Waterman (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 46; 07.09.-11.09.92 (9237)

V.R. Basili, H.D. Rombach, R.W. Selby (editors):
Experimental Software Engineering Issues, Dagstuhl-Seminar-Report; 47; 14.-18.09.92 (9238)

Y. Dittrich, H. Hastedt, P. Schefe (editors):
Computer Science and Philosophy, Dagstuhl-Seminar-Report; 48; 21.09.-25.09.92 (9239)

R.P. Daley, U. Furbach, K.P. Jantke (editors):
Analogical and Inductive Inference 1992 , Dagstuhl-Seminar-Report; 49; 05.10.-09.10.92 (9241)

E. Novak, St. Smale, J.F. Traub (editors):
Algorithms and Complexity of Continuous Problems, Dagstuhl-Seminar-Report; 50; 12.10.-16.10.92 (9242)

J. Encarnação, J. Foley (editors):
Multimedia - System Architectures and Applications, Dagstuhl-Seminar-Report; 51; 02.11.-06.11.92 (9245)

F.J. Rammig, J. Staunstrup, G. Zimmermann (editors):
Self-Timed Design, Dagstuhl-Seminar-Report; 52; 30.11.-04.12.92 (9249)