UNIVERSITÄT KOBLENZ · LANDAU

Jahrestreffen der GI-Fachgruppe 1.2.1 'Deduktionssysteme' — Kurzfassungen der Vorträge

Peter Baumgartner (Hrsg.)

23/97

Universität Koblenz-Landau
Fachbereich Informatik

Fachberichte INFORMATIK

Universität Koblenz-Landau
Institut für Informatik, Rheinau 1, D-56075 Koblenz

E-mail: researchreports@infko.uni-koblenz.de,
WWW: http://www.uni-koblenz.de/fb4/

Jahrestreffen der GI-Fachgruppe 1.2.1 „Deduktionssysteme“

Kurzfassungen der Vorträge

Schloß Dagstuhl, 30.9. – 1.10.1997

Peter Baumgartner (Hrsg.)
Universität Koblenz-Landau
Fachbereich Informatik
Rheinau 1
56075 Koblenz
Germany

peter@informatik.uni-koblenz.de

1 Vorwort

Das Jahrestreffen 1997 der GI-Fachgruppe 1.2.1 „Deduktionssysteme“ — besser bekannt als „Deduktionstreffen“ — fand vom 30.9. bis 2.10. 1997 in Schloss Dagstuhl statt. Die Organisation wurde dieses Jahr von der KI-Gruppe an der Universität Koblenz-Landau (Leitung Prof. Furbach) übernommen.

Es hatten sich 48 Teilnehmer angemeldet und es gab 19 Vorträge. Traditionsgemäß wurden in erster Linie die Arbeiten der veranstaltenden Gruppe vorgestellt (9 Vorträge), es gab aber auch reges Interesse von Seiten anderer Gruppen.

Wie bereits 1995 gab es auch diesmal wieder ein „Beweiserhappening“, das von Michael Kohlhase (Universität des Saarlandes) organisiert wurde. Es handelt sich bei dieser Veranstaltung nicht um einen Wettbewerb, sondern die Veranstaltung soll den Anwesenden die Möglichkeit geben, die verschiedenen existierenden Beweissysteme selbst in Aktion zu sehen und nach eigenen Kriterien (Geschwindigkeit, Abdeckungsgrad, Qualität der gelieferten Beweise, ...) zu vergleichen.

Peter Baumgartner
Universität Koblenz-Landau

2 Vorträge

Chandrabose Aravindan und Peter Baumgartner, Universität Koblenz
{arvind,peter}@informatik.uni-koblenz.de

A Rational and Efficient Algorithm for View Deletion in Databases

In this paper, we show how techniques from disjunctive logic programming and classical first-order theorem proving can be used for efficient (deductive) database updates. The key idea is to tranform the given database together with the update request into a disjunctive logic program and apply disjunctive techniques (such as minimal model reasoning) to solve the original update problem. We present two variants of our algorithm both of which are of polynomial space complexity. One variant, which is based on offline preprocessing, is of polynomial time complexity. We also show that both variants are rational in the sense that they satisfy certain rationality postulates stemming from philosophical works on belief dynamics.

Peter Baumgartner und Michael Kühn, Universität Koblenz
{peter,kuehn}@informatik.uni-koblenz.de

Hyper Tableaux

In [BFN96] a variant of clausal normal form tableaux called "hyper tableaux" was introduced. Hyper tableaux combine features from the tableaux world (tree structure, reading off models in special cases) with features from hyper resolution (treating variables as universally quantified).

In the talk we will report about ongoing work on improvements of the basic calculus. Its major shortcoming is the need for "purifying" substitutions in disjunctions of positive literals: in order to guarantee the soundness of the calculus, a disjunction of the form $P(x,y) \vee P(y,z)$ leads to two tableau branches containing literals $P(x,y)\pi$ and $P(y,z)\pi$, where π is a ground substitution for y. Such a technique is prohibitive if the Herbrand base explodes very quickly. In the talk we will sketch a solution to this problem which avoids the need for purifying substitutions.

Literatur

[BFN96] Baumgartner, P., U. Furbach, and I. Niemelä: 1996, 'Hyper Tableaux'. In: *Proc. JELIA 96*, Springer LNAI, 1996.

Christoph Benzmueller, AG Siekmann, Universität des Saarlandes, Saarbrücken
chris@ags.uni-sb.de

Extensionale Resolution höherer Stufe

Der Vortrag stellt den extensional vollständigen Resolutionskalkuel höherer Stufe EXTRES (Extensionale Resolution) vor und diskutiert dessen Realisierung im Beweiser LEO.

Leider bieten die Kalkuele aktueller Beweissysteme fuer Logik höherer Stufe ohne eine explizite Axiomatisierung des funktionalen Extensionalitätsprinzips (zwei Funktionen sind gleich, genau dann wenn sie auf allen Argumenten gleich sind) sowie des Extensionalitätsprinzips fuer Wahrheitswerte (Äquivalenz- und Gleichheitsrelation stimmen überein) keine Vollständigkeit in Henkinmodellen. Für viele Anwendungsdomänen (z.B. Mathematik) bilden aber gerade die Extensionalitätsprinzipien aber ein sehr wichtiges Konzept.

In Systemen behilft man sich deshalb durch eine explizite Axiomatisierung beider Prinzipien. Dies hat aber entscheidende Nachteile. Zum einen stellt sich das Problem, dass das funktionale Extensionalitaetsprinzip grundsätzlich für jeden der unendlich vielen Funktionstypen formuliert werden muss, und anderseits blähen die Extensionalitätsaxiome durch die vielen neu eingeführten flexiblen Funktions- und Prädikatsvariablen den Suchraum enorm auf.

Der extensionale Resolutionskalkül höherer Stufe EXTRES verwendet hingegen neue, negativ formulierte Regeln, die die Extensionalitätsprinzipien direkt widerspiegeln und die den Unifikationsteil und den Resolutionsteil einer Variante von Huets Higher Order Resolutionsansatz zu einem Henkinvollständigen Kalkül erweitern. Dabei sind keinerlei zusätzlichen, suchraumaufblähenden Axiome notwendig.

Der Kalkül EXTRES wurde im Beweissystem LEO (Logical Engine for Omega) aufbauend auf KEIM implementiert. LEO verwendet dazu eine leicht erweiterte Set of Support Strategie und bietet die Möglichkeit die Extensionalitätsregeln optional in die Beweissuche zu integrieren.

Sven-Erik Bornscheuer, TU Dresden
seb@inf.tu-dresden.de

Generieren Rationaler Modelle

We propose to model integrated reflective and reactive reasoning by massively parallel nonmonotonic model generation. To this end, a finite representation of models of normal logic programs is given, which is adapted to the interpretations typically produced by the meaning function $T_{\mathcal{P}}$. We lift the concepts of unification and disunification wrt this new representation. Using the resulting operators, we introduce a function $\tilde{T}_{\mathcal{P}}$, which simulates $T_{\mathcal{P}}$ in a simple and elegant way. Using $\tilde{T}_{\mathcal{P}}$ instead of $T_{\mathcal{P}}$ for generating models of completed programs, only finite representations of (possibly infinite) interpretations are processed. These representations are significantly shorter than by using related approaches. Our work overcomes typical problems of model generation and results in a tractable method, and, thereby, enables the integrated reflective and reactive reasoning we propose.

Ingo Dahn, Humboldt-Universität Berlin
dahn@mathematik.hu-berlin.de

Einsatz automatischer Theorembeweiser zur Überpruefung von Theoremen aus der Mizar-Bibliothek

Es wurde versucht, die 97 Theoreme aus dem Artikel [TS] aus der MIZAR-Bibliothek mit Hilfe von automatischen Beweisern zu verifizieren. Dazu wurden die Theoreme, die benötigten Definitionen sowie die vorhandenen Referenzen aus dem Artikel automatisch extrahiert. Die Verifikation gelang bisher für 90 Theoreme. Zwei Theoreme konnten nicht bewiesen werden, da die in [TS] angegebenen Referenzen unvollständig waren.

Für die übrigen 5 Theoreme fanden die Beweiser in der zur Verfügung stehenden Zeit von 15 Sekunden auf einer Sparc 10 Workstation keinen Beweis.

In dem Vortrag werden einige der Theoreme näher vorgestellt. Es wird erläutert, wie die Anzahl der bewiesenen Theoreme durch eine automatische Vorverarbeitung von 69 auf 90 gesteigert werden konnte. Auf die Theoreme, die nicht automatisch bewiesen werden konnten, wird genauer eingegangen.

Der behandelte Artikel ist besonders einfach. Er verwendet keine Referenzen in andere Artikel. Da in den Theoremen nur ein Typ von Objekten (nämlich Mengen) vorkommt, konnten die Beweisaufgaben direkt im Prädikatenkalkül der ersten Stufe formuliert werden.

Im zweiten Teil des Vortrags wird ein Ausblick auf weitere Beweisaufgaben gegeben. Dazu werden die mengentheoretischen Grundlagen von MIZAR (Axiomensystem von TARSKI-GROTHENDIECK) vorgestellt. Sie werden von MIZAR genutzt, um Teile des Typsystems im Objektbereich zu reflektieren und so polymorphe Typen mit Objektparametern zu realisieren.

Möglichkeiten zur Codierung des Typsystems werden kurz vorgestellt. Abschließend wird ein Ausblick auf mögliche Anwendungen automatischer Beweiser zum semantischen Retrieval von mathematischen Theoremen gegeben.

Literatur

[TS] Zinaida Trybulec and Halina 'Swiceczkowska: Boolean Properties of Sets, Journ. Form. Math. vol.1

Jürgen Dix, Universität Koblenz
dix@informatik.uni-koblenz.de

Implementing D-WFS with Theorem Proving Technology

Defining nonmonotonic semantics by using reduction operators and then using classical (minimal) entailment leads in most cases to counterintuitive semantics. This is because these operators are very technical and give rise to modifications and improvements at various places. However, such modifications might be locally well motivated, but yet have global onforeseeable effects.

We illustrate this and show how the right" reduction looks like for the recently defined disjunctive counterpart of the Wellfounded semantics. Here the original definition is given by a confluent calculus of transformations.

We show that the reduction-approach leads to an implementation in polynomial space, the first such implementation for a disjunctive semantics.

Ulrich Furbach, Universität Koblenz
uli@informatik.uni-koblenz.de

Deduktion in Koblenz

In der Arbeitsgruppe Künstliche Intelligenz laufen zur Zeit drei kooperierende Projekte, die von der Deutschen Forschungsgemeinschaft und dem Ministerium für Wirtschaft, Verkehr, Landwirtschaft und Weinbau des Landes Rheinland-Pfalz gefördert werden.

In dem DFG Deduktionsprojekt Theoriebehandlung in Beweisprozeduren für Prädikatenlogik erster Stufe werden Verfahren entwickelt, spezielle Theorien (z.B. Gleichheit) beim Theorembeweisen effizient zu behandeln.

Ebenfalls von der DFG gefördert wird das Projekt Disjunktive Logikprogrammierung. Ziel dieses Projektes ist es, logische Programmierung um disjunktive Regeln mit nicht-monotoner Negation zu erweitern.

Das Projekt Deduktive Techniken für Informations-Management-Systeme untersucht in Kooperation mit mehreren mittelständischen Unternehmen Möglichkeiten, existierende DV-Systeme mit einer deduktiven Komponente zu verbinden. Unterstützt wird dieses Projekt vom Ministerium für Wirtschaft, Verkehr, Landwirtschaft und Weinbau des Landes Rheinland-Pfalz.

Ulrich Furbach, Universität Koblenz
uli@informatik.uni-koblenz.de

Calculi for Disjunctive Logic Programming

In this paper we investigate relationships between top-down and bottom-up approaches to computation with disjunctive logic programs (DLPs). The bottom-up calculus considered, hyper tableaux, is depicted in its ground version and its relation to fixed point approaches from the literature is investigated. For the top-down calculus we use restart model elimination (RME) and show as our main result that hyper tableaux provide a bottom-up semantics for it. This generalizes the well-known result linking the T-operator to SLD-resolution for *definite* programs towards *disjunctive* programs. Furthermore we discuss that hyper tableaux can be seen as an extension of SLO-resolution.

Helmut Horacek, Universität des Saarlandes / DFKI
horacek@cs.uni-sb.de

Presenting Automatically Generated Mathematical Proofs

Adequately presenting machine-generated mathematical proofs according to the needs of a human audience is a serious challenge. One apparent difference to machine-generated proofs is that mathematical proofs as typically found in textbooks express lines of reasoning in a rather condensed form by leaving out several elementary, but logically necessary inference steps. To date, neither proof presentation techniques nor methods stemming from the area of natural language generation are able to deal with this issue in a sufficiently general and rhetorically motivated way.

In this talk, we outline our approach to base the presentation of mathematical proofs not only on solid logical, but also on well-motivated rhetorical grounds. A purely semantic, detailed, and logically complete proof tree is successively transformed into some sort of a proof skeleton, which is a much smaller but rhetorically enhanced tree. The operations include the compactification of intermediate inference steps, the substitution of a group of logical inferences by a pragmatically motivated argument, and focus-preserving regrouping of arguments. These operations are illustrated by a moderately complex example.

Dieter Hutter, DFKI GmbH
Hutter@dfki.de

Difference Reduction

Based on the success of difference reduction techniques and especially of rippling techniques in the field of inductive theorem proving we propose the use of difference reduction (rippling) and proof planning techniques also for non-inductive theorem proving. While in inductive theorem proving a proof idea is given by the task of enabling the induction hypothesis, in general theorem proving we use abstractions in order to decompose the given task of proving a theorem into different subgoals. This talk illustrates how these different techniques can be combined to an integrated approach in order to deal with equational problems.

Michael Kohlhase, Universität des Saarlandes
kohlhase@cs.uni-sb.de

Beweisplanung als Paradigma für die Integration von Computeralgebra in die Deduktion

Computeralgebrasysteme und Deduktionssysteme haben fast komplementäre Stärken und Schwächen, so können erstere sehr gut mit großen symbolischen Formeln umgehen, geben aber meist keine Begründung für die Ergebnisse. Deduktionsyssteme liefern durch ihre Beweise eine vom System unabhängige Begründung der Richtigkeit der Ergebnisse, sind aber beim symbolischen Rechnen sehr ineffizient. Es gibt daher eine Reihe von Versuchen die beiden Systemklassen zu integrieren, um Mathematiker und Ingenieure bei ihrer Arbeit zu unterstützen.

In diesem Vortrag stelle ich eine solche Integration auf der Basis des Beweisplanens vor, in der das integrierte Computeralgebrasystem Beweisinformation liefert. So kann der Anspruch des Deduktionssystems - vom System unabhängig kommunizierbare und verifizierbare Beweise zu liefern - erhalten werden.

Wenjin Lue, Uni Koblenz, Institut für Informatik
lue@informatik.uni-koblenz.de

Hyper Tableau and Minimal Model Generation

In this talk we first present the relationship between minimal models and hyper tableau of disjunctive logic program and then discuss some refinments of hyper tableau for minimal model generation. Based on this discussion we develop two minimal model generation algorithms. One is based on E-Hyper tableau, which is a variant of hyper tableau for minimal model generation and the other is based on factorization and groundedness test. And the further possible applications are also discussed.

Erica Melis, Universität Saarbrücken
melis@ags.uni-sb.de

Beweisplanen für Limes-theoreme

Beweisplanen für Limes-theoreme Proof planning provides the framework for explicitly representing methods and control knowledge in theorem proving and yields a high-level representation of proofs. In a proof planning framework, we rationally reconstruct the proofs of limit theorems for real numbers $\mathcal{R}$ that were computed by the special-purpose theorem prover and reported in BledsoeBoyer-Henneman. The proof planning framework has the advantages to provide high-level, hierarchical representations of proofs that can be expanded to checkable ND proofs and to employ explicit, modularly organized global control knowledge that allows to choose methods flexibly. Furthermore, the proof planner is general-purpose problem solver.

More generally, the solution for limit theorems demonstrates how to cope with complicated mathematical theorems in proof planning. We extend proof planning by employing explicit control-rules and by a hierarchical planning strategy. Furthermore we combine proof planning with constraint solving. Experiments show the influence of these additional mechanisms on the performance of proof planning. The proofs of LIM+, LIM*, and CONTINOUS have been automatically proof planned in the extended proof planner OMEGA.

As a knowledge engineering result, the paper presents appropriate constraint solving, methods, and control knowledge that restrict the search and reflect human ways of theorem proving.

David von Oheimb, Fakultät für Informatik, TU München
`oheimb@informatik.tu-muenchen.de`

Java_light is Type-safe - Definitely

Java_light is a large subset of the sequential part of Java. We discuss its type system, the operational semantics, and a proof of type soundness. All definitions and proofs have been done formally in the theorem prover Isabelle/HOL. Thus we demonstrate that machine-checking the design of non-trivial programming languages has become a reality.

Uwe Petermann, FB IMN, HTWK Leipzig
`uwe@imn.htwk-leipzig.de`

Einbau von Hybridtheorien

Hybrid reasoning is understood as the cooperation of a foreground reasoner with a background reasoner. The foreground reasoner takes care of the general logical structure of a formula to be proved or refuted. The background reasoner is consulted whenever the meaning of special built-ins has to be considered. A number of general results forms a framework for building-in theories into theorem proving calculi.

Reasoning in hybrid theory means that the background reasoner is considered as a hybrid itself. This approach is motivated by applications, among them the algebraic translation of multi-modal logic into a fragment of first-order logic. Reasoning in the target logic means reasoning in a theory consisting of two sub-theories. In this example one sub-theory is a definite theory the other one is an equational theory. Also reasoning in a constraint theory may be seen as reasoning in a hybrid theory.

The paper formulates conditions being sufficient for constructing a complete calculus from complete calculi for the constituents of a given hybrid theory. The concept of a set of theory connections which is complete with respect to a query language and has a solvable unification problem has been applied for this result.

Frieder Stolzenburg und Oliver Obst
{stolzen,frvit}@informatik.uni-koblenz.de

Reasoning with Constraints and Well-Founded Negation

Constraint reasoning and logic programming have been combined with great success. Constraint logic programming (CLP) [5] has gained a lot of interest, since it combines both fields in a theoretically sound manner while achieving efficiency by dedicated constraint solvers for practical applications. However in CLP, only Horn clauses are considered, i.e. rules of the form $A \leftarrow B_1 \wedge \cdots \wedge B_n$. But in many cases it is desirable also to have some form of negation in logic programming—see [1] for a survey—, which allows us to express exceptions to rules (e.g. with default rules) and to shorten formalisations (e.g. with the negation as finite failure rule) among other things. This is especially useful for solving diagnosis problems.

Therefore, we here propose a combination of constraint reasoning and general logic programs, where negation and also disjunctive rules, i.e. rules with more than one literal in their heads, are allowed, while preserving the full power of constraint reasoning. We do this on the basis of the disjunctive well-founded semantics (D-WFS) [2], since it avoids many of the problems that other semantics have. It is sound for general logic programs and can be adapted to the non-ground case with variables. In addition, the relevance property holds for this semantics, i.e. the truth value of an atom is determined solely by the part of the program that atom depends on. The D-WFS coincides with the well-founded semantics [8] for normal programs (i.e. without disjunctive rules) and with the generalised closed world asumption [7] for positive disjunctive programs (i.e. without non-monotonic negation).

Nevertheless, nearly all approaches assume "without loss of generality"—as most authors put it—that the considered programs are grounded, i.e. they do not contain variables. However, we think that one of the most important advantages of the logic programming paradigm and therefore the success of Prolog is its ability to compute answer-substitutions for a given query with variables. Although semantics for logic programs with negation are undecidable, if function symbols and variables are allowed, we are convinced that query answering mechanisms for the non-ground case have great advantages over the propositional case. Of course, such procedures can only be sound and not complete.

Our approach is essentially based on a calculus of program transformations that has been recently shown to be confluent and terminating for ground programs [2]. The most important transformation in this calculus is the partial evaluation property (GPPE) adapted for disjunctive programs. Unfortunately, GPPE is not sound for rules with variables because of the occurrence of unifiable atoms in the heads of rules.

We make the GPPE sound by introducing equational constraints [6, 3]. This immediately leads us to introduce constraint disjunctive logic programs and consequently to extend our transformations to this class of programs. In principle, any constraint theory known from CLP can be exploited in the context of non-monotonic reasoning, not only equational constraints over the Herbrand domain. However, the respective constraint solver must be able to treat negative constraints of the considered constraint domain. Surprisingly, this framework shares the same nice properties as the original calculus. In summary, our framework—which is explained in detail in [4]—is a general combination of the two paradigms *constraint logic programming* and *non-monotonic reasoning*.

In this context, we want to present the following: First, we will outline the new framework, called *constraint D-WFS*. After that, we will show the usefulness of the approach with a practical example from diagnosis. Finally, we will report on the implementation of our calculus, which aims at incorporating a general logic programming deduction system with constraints.

Literatur

[1] K. R. Apt and R. N. Bol. Logic programming and negation: A survey. *Journal of Logic Programming*, 19/20:9–71, 1994.

[2] S. Brass and J. Dix. A general approach to bottom-up computation of disjunctive semantics. In J. Dix, L. M. Pereira, and T. C. Przymusinski, editors, *Selected Papers of the Workshop on Non-Monotonic Extensions of Logic Programming in Conjunction with 11th International Conference of Logic Programming 1994*, pages 127–155, Santa Margherita, 1995. Springer, Berlin, Heidelberg, New York. LNAI 927.

[3] H. Comon and P. Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7:371–425, 1989.

[4] J. Dix and F. Stolzenburg. A framework to incorporate non-monotonicity into constraint logic programming. Fachberichte Informatik 16/97, Universität Koblenz, 1997. To appear in *Journal of Logic Programming*.

[5] J. Jaffar and M. J. Maher. Constraint logic programming: a survey. *Journal of Logic Programming*, 19/20:503–581, 1994.

[6] M. J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, pages 348–359. Computer Society Press, 1988.

[7] J. Minker. On indefinite databases and the closed world assumption. In *Proceedings of the 6th Conference on Automated Deduction*, pages 292–308, New York, 1982. Springer, Berlin, Heidelberg, New York.

[8] A. van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38:620–650, 1991.

Bernd Thomas, Uni Koblenz, Institut für Informatik
bthomas@informatik.uni-koblenz.de

GLUE: Heterogeneous Sources of Information in a Logic Programming System

There is a vast number of different data sources computer systems work with. One aim of Information Management Systems is to provide a tool that can reason about information from heterogeneous data sources, by keeping all these data as it is but provide means to integrate them in one framework.

We consider a logic programming engine—or an automated theorem prover—to be the proper tool for the kernel of a system which tries to combine several sources of external information. Mathematical logic for classical deduction has a long tradition and is well understood. Furthermore it provides an extension of pure Horn clauses. Thus we have decided to use it as a basis for further development.

The integration of external, heterogeneous sources of information into a deductive system bears several problems to be solved. The external sources have to be made accessible. A common representation has to be used if possible. Otherwise means have to be provided to take into consideration the semantics of the external sources of information. Finally, the performance has to be brought into an acceptable region. These issues are discussed within the context of the system GLUE.

3 Systeme beim Beweiserhappening

Die Fachgruppe 1.2.1, Deduktion, der GI veranstaltete am Donnerstag, 2. Oktober 1997 ein „Beweiserhappening" im Rahmen ihres Jahrestreffen 1997.

Es handelt sich bei dieser Veranstaltung im Gegensatz zu der CADE System Competition nicht um einen Wettbewerb, sondern die Veranstaltung soll den Anwesenden die Möglichkeit geben, die verschiedenen existierenden Beweissysteme selbst in Aktion zu sehen und nach eigenen Kriterien (Geschwindigkeit, Abdeckungsgrad, Qualität der gelieferten Beweise, ...) zu vergleichen.

Es wurden Aufgaben aus den folgenden Anwendungskategorien gestellt: Diagnose, Elementare Algebra, Gleichheit, Planung, Programmverifikation, Software Wiederverwendung, Translated Modal Logics.

Zum Beweiserhappening wurden die folgenden Systeme angemeldet.

Bernhard Beckert, Universität Karlsruhe
beckert@ira.uka.de

3TAP

$_3T^AP$ is a tableau-based theorem prover for many-valued first-order logics with sorts (in the two-valued version with equality); it is implemented in Prolog.

$_3T^AP$ is able to handle full first-order logics with any finite number of truth values. Hierarchical (tree-shaped) sorts attached to terms are supported. In the two-valued case, special handling of the equality predicate is provided. It is possible either to prove a theorem or to try to check the consistency of the axioms.

$_3T^AP$'s calculus is based on free variable semantic tableaux (with Skolem functions).

$_3T^AP$ is available via the *World Wide Web* and via anonymous ftp. To obtain further information or to download the source code and the user's manual, use the $_3T^AP$ home page `http://i12www.ira.uka.de/~threetap/`. Alternatively, connect via anonymous ftp to `sonja.ira.uka.de` (`129.13.31.3`) and change to the directory `pub/threetap`.

Christoph Benzmueller, AG Siekmann, Universität des Saarlandes, Saarbrücken
chris@ags.uni-sb.de

LEO - Logical Engine for Omega

LEO ist ein extensional vollständiger Resolutionsbeweiser für Logik höherer Stufe. (Siehe auch obige Kurzfassung zum Vortrag Extensionale Resolution auf höherer Stufe)

Dieter Hutter, DFKI GmbH
Hutter@dfki.de

INKA

The INKA system is a first-order theorem prover with induction based on the explicit induction paradigm. In contrast to Nqthm, the Boyer-Moore prover, the system is based on a full first-order calculus, a special variant of an order-sorted resolution calculus with paramodulation. However, it is not specialized on inductive proofs but possesses a powerful predicate-logic proof component.

INKA is designed to be used for practical applications of inductive theorem proving, for instance, in the area of formal software development where a lot of deductive support for verification tasks is necessary. In that area the occuring deduction problems *have to be* solved in order to guarantee the correctness of the resulting software. Thus, a theorem prover must allow the user to interact, and, hence, in addition to completely automated proofs we follow the paradigm of strategic, interactive proof support.

Dorothea Schäfer, Universität Koblenz
doro@informatik.uni-koblenz.de

PROTEIN

PROTEIN (*PRO*ver with a *T*heory *E*xtension *IN*terface) is a PTTP-based first order theorem prover over built-in theories. Besides various standard-refinements known for model elimination, PROTEIN also offers the restart model elimination as a variant of model elimination which does not need contrapositives.

PROTEIN comes with tools for theory handling (automatic recognition of theories where precompiled theory reasoners are available in a library, and a tool to transform a Horn theory into a theory reasoner).

PROTEIN also includes features for minimal model reasoning which are required for disjunctive logic programming.

PROTEIN was developed at the University of Koblenz in the course of the DFG-research programme "Deduction". Please see the URL
http://www.uni-koblenz.de/ag-ki/Implementierungen/Protein/ for further information.

Gerhard Schellhorn, Universität Ulm
schellhorn@informatik.uni-ulm.de

KIV

KIV is ein fortgeschrittenes Werkzeug zur Enwicklung korrekter Software. Es unterstützt:

- Hierarchische, formale Spezifikation von Software Designs
- Spezifikation von Sicherheitsmodellen
- Nachweis von Spezifikationseigenschaften
- Modulare Implementierung von Spezifikationskomponenten
- Modulare Verifikation von Implementierungen
- Inkrementelle Verification und Fehlerbehebung
- Wiederverwendung von Spezifikationen, Beweisen, und verifizierten Komponenten

Nähere Informationen zu KIV finden sich auf der WWW-Homepage des KIV-Systems in Ulm: *http://www.informatik.uni-ulm.de/pm/kiv/kiv.html*.

Christoph Weidenbach, Max-Planck-Institut für Informatik
weidenb@mpi-sb.mpg.de

SPASS

SPASS = Superposition + PA + Sorts + Splitting

Available Research Reports (since 1995):

1997

23/97 *Peter Baumgartner (Hrsg.).* Jahrestreffen der GI-Fachgruppe 1.2.1 'Deduktionssysteme' — Kurzfassungen der Vorträge.

22/97 *Jens M. Felderhoff, Thomas Marx.* Erkennung semantischer Integritätsbedingungen in Datenbankanwendungen.

21/97 *Angelika Franzke.* Specifying Object Oriented Systems using GDMO, ZEST and SDL'92.

20/97 *Angelika Franzke.* Recommendations for an Improvement of GDMO.

19/97 *Jürgen Dix, Luís Moniz Pereira, Teodor Przymusinski.* Logic Programming and Knowledge Representation (LPKR '97) (Proceedings of the ILPS '97 Postconference Workshop).

18/97 *Lutz Priese, Harro Wimmel.* A Uniform Approach to True-Concurrency and Interleaving Semantics for Petri Nets.

17/97 *Ulrich Furbach (Ed.).* IJCAI-97 Workshop on Model Based Automated Reasoning.

16/97 *Jürgen Dix, Frieder Stolzenburg.* A Framework to Incorporate Non-Monotonic Reasoning into Constraint Logic Programming.

15/97 *Carlo Simon, Hanno Ridder, Thomas Marx.* The Petri Net Tools Neptun and Poseidon.

14/97 *Juha-Pekka Tolvanen, Andreas Winter (Eds.).* CAiSE'97 — 4th Doctoral Consortium on Advanced Information Systems Engineering, Barcelona, June 16-17, 1997, Proceedings.

13/97 *Jürgen Ebert, Roger Süttenbach.* An OMT Metamodel.

12/97 *Stefan Brass, Jürgen Dix, Teodor Przymusinski.* Super Logic Programs.

11/97 *Jürgen Dix, Mauricio Osorio.* Towards Well-Behaved Semantics Suitable for Aggregation.

10/97 *Chandrabose Aravindan, Peter Baumgartner.* A Rational and Efficient Algorithm for View Deletion in Databases.

9/97 *Wolfgang Albrecht, Dieter Zöbel.* Integrating Fixed Priority and Static Scheduling to Maintain External Consistency.

8/97 *Jürgen Ebert, Alexander Fronk.* Operational Semantics of Visual Notations.

7/97 *Thomas Marx.* APRIL - Visualisierung der Anforderungen.

6/97 *Jürgen Ebert, Manfred Kamp, Andreas Winter.* A Generic System to Support Multi-Level Understanding of Heterogeneous Software.

5/97 *Roger Süttenbach, Jürgen Ebert.* A Booch Metamodel.

4/97 *Jürgen Dix, Luis Pereira, Teodor Przymusinski.* Prolegomena to Logic Programming for Non-Monotonic Reasoning.

3/97 *Angelika Franzke.* GRAL 2.0: A Reference Manual.

2/97 *Ulrich Furbach.* A View to Automated Reasoning in Artificial Intelligence.

1/97 *Chandrabose Aravindan, Jürgen Dix, Ilkka Niemelä .* DisLoP: A Research Project on Disjunctive Logic Programming.

1996

28/96 *Wolfgang Albrecht.* Echtzeitplanung für Alters- oder Reaktionszeitanforderungen.

27/96 *Kurt Lautenbach.* Action Logical Correctness Proving.

26/96 *Frieder Stolzenburg, Stephan Höhne, Ulrich Koch, Martin Volk.* Constraint Logic Programming for Computational Linguistics.

25/96 *Kurt Lautenbach, Hanno Ridder.* Die Lineare Algebra der Verklemmungsvermeidung — Ein Petri-Netz-Ansatz.

24/96 *Peter Baumgartner, Ulrich Furbach.* Refinements for Restart Model Elimination.

23/96 *Peter Baumgartner, Peter Fröhlich, Ulrich Furbach, Wolfgang Nejdl.* Tableaux for Diagnosis Applications.

22/96 *Jürgen Ebert, Roger Süttenbach, Ingar Uhe.* Meta-CASE in Practice: a Case for KOGGE.

21/96 *Harro Wimmel, Lutz Priese.* Algebraic Characterization of Petri Net Pomset Semantics.

20/96 *Wenjin Lu.* Minimal Model Generation Based on E-Hyper Tableaux.

19/96 *Frieder Stolzenburg.* A Flexible System for Constraint Disjunctive Logic Programming.

18/96 *Ilkka Niemelä (Ed.).* Proceedings of the ECAI'96 Workshop on Integrating Nonmonotonicity into Automated Reasoning Systems.

17/96 *Jürgen Dix, Luis Moniz Pereira, Teodor Przymusinski.* Non-monotonic Extensions of Logic Programming: Theory, Implementation and Applications (Proceedings of the JICSLP '96 Postconference Workshop W1).

16/96 *Chandrabose Aravindan.* DisLoP: A Disjunctive Logic Programming System Based on PROTEIN Theorem Prover.

15/96 *Jürgen Dix, Gerhard Brewka.* Knowledge Representation with Logic Programs.

14/96 *Harro Wimmel, Lutz Priese.* An Application of Compositional Petri Net Semantics.

13/96 *Peter Baumgartner, Ulrich Furbach.* Calculi for Disjunctive Logic Programming.

12/96 *Klaus Zitzmann.* Physically Based Volume Rendering of Gaseous Objects.

11/96 *J. Ebert, A. Winter, P. Dahm, A. Franzke, R. Süttenbach.* Graph Based Modeling and Implementation with EER/GRAL.

10/96 *Angelika Franzke.* Querying Graph Structures with G^2QL.

9/96 *Chandrabose Aravindan.* An abductive framework for negation in disjunctive logic programming.

8/96 *Peter Baumgartner, Ulrich Furbach, Ilkka Niemelä .* Hyper Tableaux.

7/96 *Ilkka Niemelä, Patrik Simons.* Efficient Implementation of the Well-founded and Stable Model Semantics.

6/96 *Ilkka Niemelä .* Implementing Circumscription Using a Tableau Method.

5/96 *Ilkka Niemelä .* A Tableau Calculus for Minimal Model Reasoning.

4/96 *Stefan Brass, Jürgen Dix, Teodor. C. Przymusinski.* Characterizations and Implementation of Static Semantics of Disjunctive Programs.

3/96 *Jürgen Ebert, Manfred Kamp, Andreas Winter.* Generic Support for Understanding Heterogeneous Software.

2/96 *Stefan Brass, Jürgen Dix, Ilkka Niemelä, Teodor. C. Przymusinski.* A Comparison of STATIC Semantics with D-WFS.

1/96 *J. Ebert (Hrsg.).* Alternative Konzepte für Sprachen und Rechner, Bad Honnef 1995.

1995

21/95 *J. Dix and U. Furbach.* Logisches Programmieren mit Negation und Disjunktion.

20/95 *L. Priese, H. Wimmel.* On Some Compositional Petri Net Semantics.

19/95 *J. Ebert, G. Engels.* Specification of Object Life Cycle Definitions.

18/95 *J. Dix, D. Gottlob, V. Marek.* Reducing Disjunctive to Non-Disjunctive Semantics by Shift-Operations.

17/95 *P. Baumgartner, J. Dix, U. Furbach, D. Schäfer, F. Stolzenburg.* Deduktion und Logisches Programmieren.

16/95 *Doris Nolte, Lutz Priese.* Abstract Fairness and Semantics.

15/95 *Volker Rehrmann (Hrsg.).* 1. Workshop Farbbildverarbeitung.

14/95 *Frieder Stolzenburg, Bernd Thomas.* Analysing Rule Sets for the Calculation of Banking Fees by a Theorem Prover with Constraints.

13/95 *Frieder Stolzenburg.* Membership-Constraints and Complexity in Logic Programming with Sets.

12/95 *Stefan Brass, Jürgen Dix.* D-WFS: A Confluent Calculus and an Equivalent Characterization..

11/95 *Thomas Marx.* NetCASE — A Petri Net based Method for Database Application Design and Generation.

10/95 *Kurt Lautenbach, Hanno Ridder.* A Completion of the S-invariance Technique by means of Fixed Point Algorithms.

9/95 *Christian Fahrner, Thomas Marx, Stephan Philippi.* Integration of Integrity Constraints into Object-Oriented Database Schema according to ODMG-93.

8/95 *Christoph Steigner, Andreas Weihrauch.* Modelling Timeouts in Protocol Design..

7/95 *Jürgen Ebert, Gottfried Vossen.* I-Serializability: Generalized Correctness for Transaction-Based Environments.

6/95 *P. Baumgartner, S. Brüning.* A Disjunctive Positive Refinement of Model Elimination and its Application to Subsumption Deletion.

5/95 *P. Baumgartner, J. Schumann.* Implementing Restart Model Elimination and Theory Model Elimination on top of SETHEO.

4/95 *Lutz Priese, Jens Klieber, Raimund Lakmann, Volker Rehrmann, Rainer Schian.* Echtzeit-Verkehrszeichenerkennung mit dem Color Structure Code — Ein Projektbericht.

3/95 *Lutz Priese.* A Class of Fully Abstract Semantics for Petri-Nets.

2/95 *P. Baumgartner, R. Hähnle, J. Posegga (Hrsg.).* 4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods — Poster Session and Short Papers.

1/95 *P. Baumgartner, U. Furbach, F. Stolzenburg.* Model Elimination, Logic Programming and Computing Answers.