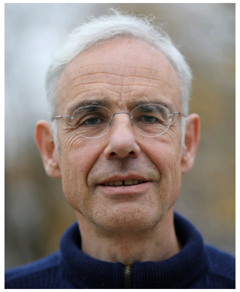
Herbrand Award 2020
Acceptance Speech

Franz Baader

Deutsche Forschungsgemeinschaft

TECHNISCHE UNIVERSITÄT DRESDEN

RWTH AACHEN UNIVERSITY



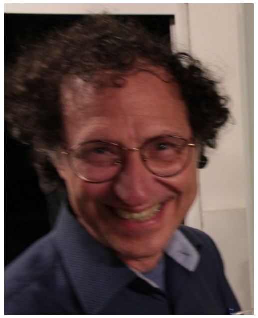






Herbrand Award
Andrei Voronkov







Herbrand Award
Edmund M. Clarke













Spot the winners!


Unification
First Workshop
in Val-d´Ajol
1987
© Pierre Lescanne

Unification modulo theories

Treat certain properties of function symbols not by adding equational axioms to the first-order theory, but by special unification algorithms.

[Plotkin, 1972]

AC [Peterson, Stickel; 1981]

[McCune; 1997]

most general unifier $\Rightarrow$ complete set of unifier

Complications:

- complete sets may be infinite, A
- minimal complete sets may not even exist, AI

Dresden

Journal of Automated Reasoning **2** (1986) 277–281.

Unification under Associativity and Idempotenc of Type Nullary★

MANFRED SCHMIDT-SCHAUSS
Universität Kaiserslautern; 6750 Kaiserslautern, F.R. Germany
(UUCP: seismo! unido! uklirb! schauss)

(Received 31 January 1986)

Abstract. It is shown, that there exist a unification problem $\langle s = t\rangle_{AI}$, for which the set of solution associativity and idempotence is not empty. But $\mu U\Sigma_{AI}(s, t)$, the complete and minimal subset of of solutions does not exist, i.e. $A + I$ is of type nullary. This is the first known standard first orde with this unpleasant feature.

Key words. Unification, Equational Theories, Idempotent Semigroups

1. Introduction

Unification theory is concerned with the problem to find solutions for an eq $\langle s = t\rangle$, where s and t are terms. Solutions of $\langle s = t\rangle$ are substitutions $\sigma s = \sigma t$. The substitution σ is called a *unifier* for s and t.

An extension of this problem is the T-unification problem: Given a set of equ T we say two terms t_1 and t_2 are equal w.r.t. T, denoted as $t_1 =_T t_2$, iff $t_1 = t_2$ lo follows from T. A T-unification problem $\langle s = t\rangle_T$ is the problem to find solut such that $\sigma s =_T \sigma t$.

The set of all unifying substitutions (i.e. of all solutions) of $\langle s = t\rangle_T$ is denc $U\Sigma_T(s, t)$. In many cases, the set of all solutions $U\Sigma_T(s, t)$ can be generated f minimal subset of solutions, the set of general unifiers $\mu U\Sigma_T(s, t)$, which is defi follows: We say the substitution σ is more general than τ on the set of variat $(\tau \leqslant_T \sigma[\![W]\!])$ iff there exists a substitution λ, such that $\tau x =_T \lambda\sigma x$ for all $x \in V$ set $\mu U\Sigma_T(s, t)$ is characterized by three conditions:

(i) correctness: $\mu U\Sigma_T(s, t) \subseteq U\Sigma_T(s, t)$
(ii) completeness: $\forall\theta \in U\Sigma_T(s, t)\exists\sigma \in \mu U\Sigma_T(s, t)\ \theta \leqslant_T \sigma[\![W]\!]$ where $W =$
(iii) minimality: $\forall\sigma, \tau \in \mu U\Sigma_T(s, t)\ \sigma \leqslant_T \tau[\![W]\!] \Rightarrow \sigma = \tau.$ where $W =$

Journal of Automated Reasoning **2** (1986) 283–286.

The Theory of Idempotent Semigroups is of Unification Type Zero

FRANZ BAADER
Institut für Mathematische Maschinen und Datenverarbeitung I Martensstraße 3, 8250 Erlangen, West Germany

(Received 20 May 1986)

1. E-Unification

Let E be a set of equations and $=_E$ the equality of terms induced by E. A substitutio θ is called an *E-unifier* for the pair of terms s, t iff $s\theta =_E t\theta$. The set of all E-unifier for s and t is denoted by $U\Sigma_E(s, t)$.

We define a quasi-ordering $\leqq_E$ on $U\Sigma_E(s, t)$ by

$\theta_1 \leqq_E \theta_2$: $\leftrightarrow$ There exists a substitution λ satisfying $x\theta_1 =_E x\theta_2\lambda$ for all varia bles x occurring in s or t.

In this case θ_1 is called an instance of θ_2.

We write $\theta_1 =_E \theta_2$ iff $x\theta_1 =_E x\theta_2$ for all variables x occurring in s or t.

A *set of most general E-unifiers* $\mu U\Sigma_E(s, t)$ for the unification problem $\langle s =_E t\rangle$ i defined as

(1) $\mu U\Sigma_E \subseteq U\Sigma_E$.
(2) For all $\theta \in U\Sigma_E$ there exists a $\sigma \in \mu U\Sigma_E$ such that $\theta \leqq_E \sigma$.
(3) For all θ_1, θ_2 in $\mu U\Sigma_E$ $\theta_1 \leqq_E \theta_2$ implies $\theta_1 = \theta_2$.

Equational theories may be classified according to the cardinality or the existenc of $\mu U\Sigma_E$ as follows:

(1) If $\mu U\Sigma_E(s, t)$ exists for all terms s, t and has at most one element then E is calle *unitary*.
(2) If $\mu U\Sigma_E(s, t)$ exists for all terms s, t and has finite cardinality then E is calle *finitary*.
(3) If $\mu U\Sigma_E(s, t)$ exists for all terms s, t and for some terms u, v $\mu U\Sigma_E(u, v)$ is infinit then E is called *infinitary*.
(4) If for some terms s, t $\mu U\Sigma_E(s, t)$ does not exist then E is said to be of *unificatio type zero*.

In this paper it will be shown that the theory of idempotent semigroup $AI = \{(xy)z = x(yz), x^2 = x\}$ is of unification type zero. This seems to be the firs natural example of a first order theory of this type which is not an artificial constructio as in [3].

Unification modulo theories

Treat certain properties of function symbols not by adding equational axioms to the first-order theory, but by special unification algorithms.

[Plotkin, 1972]

AC [Peterson, Stickel; 1981]

[McCune; 1997]

most general unifier $\Rightarrow$ complete set of unifier

Complications:

- complete sets may be infinite, A
- minimal complete sets may not even exist, AI
- complete sets may be finite, but quite large, AC

  [Kapur, Narendran; LICS 1992]
  Double-exponential complexity of computing a complete set of AC-unifiers

Unification modulo theories

Treat certain properties of function symbols not by adding equational axioms to the first-order theory, but by special unification algorithms.

[Plotkin, 1972]

AC [Peterson, Stickel; 1981]

[McCune; 1997]

most general unifier $\Rightarrow$ complete set of unifier

Complications:

- complete sets may be infinite, A
- minimal complete sets may not even exist, AI
- complete sets may be finite, but quite large, AC
- the combination problem.

[Stickel; 1981] AC

[Fages; 1984]

[Schmidt-Schauß; 1989] general

Dresden

Unification modulo theories

Treat certain properties of function symbols not by adding equational axioms to the first-order theory, but by special unification algorithms.

[Plotkin, 1972]

AC [Peterson, Stickel; 1981]

[McCune; 1997]

most general unifier $\Rightarrow$ complete set of unifier

Complications:

- ~~complete sets may be infinite,~~ ~~A~~
- ~~minimal complete sets may not even exist,~~ AI
- ~~complete sets may be finite, but quite large,~~ ~~AC~~
- the combination problem. [Stickel; 1981] [Fages; 1984] AC [Schmidt-Schauß; 1989] general

unifiability constraints

$\Downarrow$

new challenge

Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures

Franz Baader
DFKI
Stuhlsatzenhausweg 3
6600 Saarbrücken 11, Germany
e-mail: baader@dfki.uni-sb.de

Klaus U. Schulz
CIS
University Munich, Leopoldstr. 139
8000 München 40, Germany
e-mail: schulz@cis.uni-muenchen.dbp.de

Abstract

Most of the work on the combination of unification algorithms for the union of disjoint equational theories has been restricted to algorithms which compute finite complete sets of unifiers. Thus the developed combination methods usually cannot be used to combine decision procedures, i.e., algorithms which just decide solvability of unification problems without computing unifiers. In this paper we describe a combination algorithm for decision procedures which works for arbitrary equational theories, provided that solvability of so-called unification problems with constant restrictions—a slight generalization of unification problems with constants—is decidable for these theories. As a consequence of this new method, we can for example show that general A-unifiability, i.e., solvability of A-unification problems with free function symbols, is decidable. Here A stands for the equational theory of one associative function symbol.

Our method can also be used to combine algorithms which compute finite complete sets of unifiers. Manfred Schmidt-Schauß' combination result, the until now most general result in this direction, can be obtained as a consequence of this fact. We also get the new result that unification in the union of disjoint equational theories is finitary, if general unification—i.e., unification of terms with additional free function symbols—is finitary in the single theories.

CADE 1992
UNIF 1991

Difference to
Nelson-Oppen combination:

Solvability of equations in the E-free algebra — Unification

versus

Solvability of equations in some model of E — Nelson-Oppen

J. Symbolic Computation (1996) **21**, 211–243

Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures

FRANZ BAADER[†] AND KLAUS U. SCHULZ[‡]

[†] *LuFg Theoretical Computer Science, RWTH Aachen, Ahornstr.55, 52074 Aachen, Germany*

[‡] *CIS, University of Munich, Wagmüllerstr.23,80538 München, Germany*

(Received 15 November 1993)

Most of the work on the combination of unification algorithms for the union of disjoint equational theories has been restricted to algorithms that compute finite complete sets of unifiers. Thus the developed combination methods usually cannot be used to combine decision procedures, i.e., algorithms that just decide solvability of unification problems without computing unifiers. In this paper we describe a combination algorithm for decision procedures that works for arbitrary equational theories, provided that solvability of so-called unification problems with constant restrictions—a slight generalization of unification problems with constants—is decidable for these theories. As a consequence of this new method, we can, for example, show that general A-unifiability, i.e., solvability of A-unification problems with free function symbols, is decidable. Here A stands for the equational theory of one associative function symbol.

Our method can also be used to combine algorithms that compute finite complete sets of unifiers. Manfred Schmidt-Schauß' combination result, the until now most general result in this direction, can be obtained as a consequence of this fact. We also obtain the new result that unification in the union of disjoint equational theories is finitary, if general unification—i.e., unification of terms with additional free function symbols—is finitary in the single theories.

Dresden

Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures

FRANZ BAADER[†] AND KLAUS U. SCHULZ[‡]

The combination algorithm applies

- several nondeterministic polynomial guessing steps to
- produce unification problems in the component theories
- that can be solved separately.

with linear constant restrictions

The following statements are equivalent for an equational theory E:

- E-unification with additional free function symbols is decidable.
- E-unification with linear constant restrictions is decidable.
- The positive theory of E is decidable.

Going from E-unification (with constants) to E-unification with linear constant restrictions can increase the complexity considerably, and may even cause undecidability.

Dresden

Information Processing Letters 67 (1998) 215–220

On the complexity of Boolean unification

Franz Baader *, 1

LuFg Theoretical Computer Science, RWTH Aachen, Ahornstraße 55, 52074 Aachen, Germany

Received 11 June 1997
Communicated by H. Ganzinger

Abstract

Unification modulo the theory of Boolean algebras has been investigated by several autors. Nevertheless, the exact complexity of the decision problem for unification with constants and general unification was not known. In this research note, we show that the decision problem is Π_2^p-complete for unification with constants and PSPACE-complete for general unification. In contrast, the decision problem for elementary unification (where the terms to be unified contain only symbols of the signature of Boolean algebras) is "only" NP-complete. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Computational complexity; Automatic theorem proving

1. Introduction

Boolean unification, i.e., unification modulo the theory of Boolean algebras or rings, has been considered by several authors [5,15,14]. On the one hand, this problem is of interest for research in unification theory since, unlike theories such as associativity-commutativity, the theory of Boolean algebras is unitary even for unification with constants (where the terms to be unified may contain additional free constant symbols). In addition, well-known results from mathematics [2,13,17] can be used to compute the most general unifier of a given (solvable) unification problem. General Boolean unification (where the terms to be unified may contain additional free function symbols) is still finitary, but no longer unitary [18]. From a practical point of view, a Prolog system enhanced by Boolean unification can, e.g., be used to support hardware verification and design tasks [5,19].

The emphasis in the work on Boolean unification was on developing algorithms that compute a most general unifier for unification problems with constants [5,15,14], or finite complete sets of unifiers for general unification problems [18,3]. Of course, such algorithms can also be used to decide solvability of a given unification problem. However, the complexity of a decision procedure obtained this way need not be optimal. In fact, to the best of our knowledge, the exact complexity of the decision problem for Boolean unification has only been proved for elementary unification, where it is easily seen to be NP-complete. For unification with constants, Π_2^p complexity is mentioned (without a complete proof) in [10].

In this research note, we will determine the complexity of the decision problem for the following kinds of Boolean unification problems: unification problems with constants, unification problems with linear constant restrictions (which were introduced in the context of combination of unification algorithms [1]), and general unification problems. To be more precise, we

* Partially supported by the EC Working Group CCL II.
[1] Email: baader@informatik.rwth-aachen.de.

PII: S0020-0190(98)00106-9

J Autom Reasoning (2012) 48:363–390
DOI 10.1007/s10817-010-9201-2

E-unification with Constants vs. General E-unification

Jan Otop

Received: 8 November 2008 / Accepted: 8 August 2010 / Published online: 21 August 2010

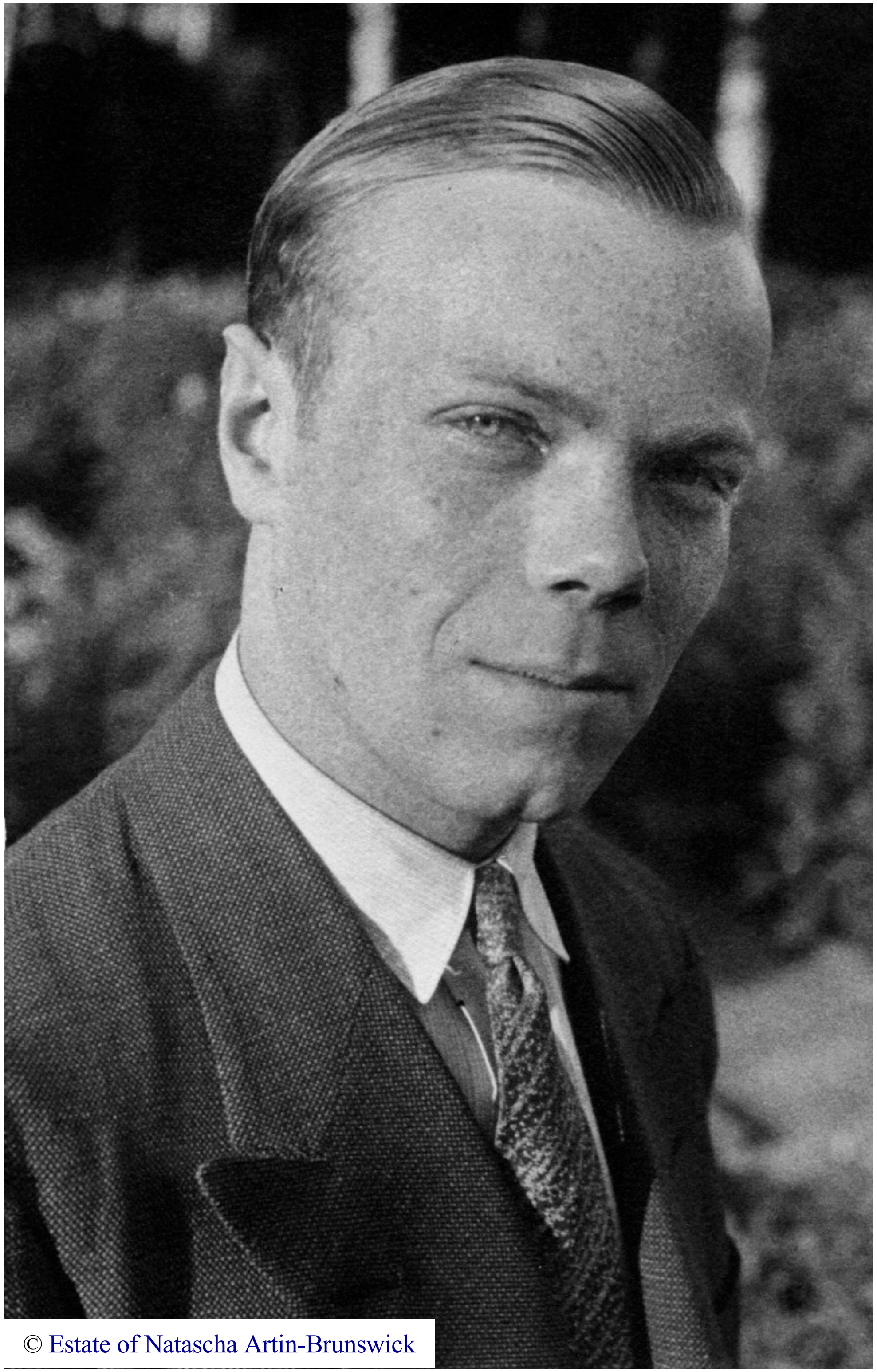
Abstract We present a solution to Problem #66 from the RTA open problem list. The question is whether there exists an equational theory E such that E-unification with constants is decidable but general E-unification is undecidable. The answer is positive and we show such a theory. The problem has several equivalent formulations, therefore the solution has many consequences. Our result also shows, that there exist two theories E_1 and E_2 over disjoint signatures, such that E_1-unification with constants and E_2-unification with constants are decidable, but $(E_1 \cup E_2)$-unification with constants is undecidable.

Keywords E-unification with constants · General E-unification · Combination problem

1 Introduction

The aim of the combination problem for unification is to find a procedure which using an E_1-unification algorithm and an E_2-unification algorithm constructs an $(E_1 \cup E_2)$-unification algorithm. The combination problem was intensively studied by many researchers. The main question is which theories admit a combination procedure. Many results were published for particular types of theories (simple, regular and collapse free, etc.). These results are summarized in [6]. Schmidt-Schauss presented a more general result in [12]. He has not restricted theories to have any particular type, instead he showed that all equational theories E_1, E_2 over disjoint signatures that have decidable constant elimination problems admit a combination procedure. This result was improved by Baader and Schulz in [3, 4]. They showed that all equational theories E_1, E_2 over disjoint signatures having decidable E_1- and E_2-unification

J. Otop (✉)
Institute of Computer Science, University of Wrocław,
ul. Joliot-Curie 15, 50-383, Wrocław, Poland
e-mail: jotop@cs.uni.wroc.pl



Jacques Herbrand

sketched a unification algorithm akin to the transformation-based algorithm by Martelli-Montanari.

In the considered setting without Skolemization, he would have actually needed linear constant restrictions to express the quantifier prefix.

https://www.mathouriste.eu/Herbrand/Herbrand.html

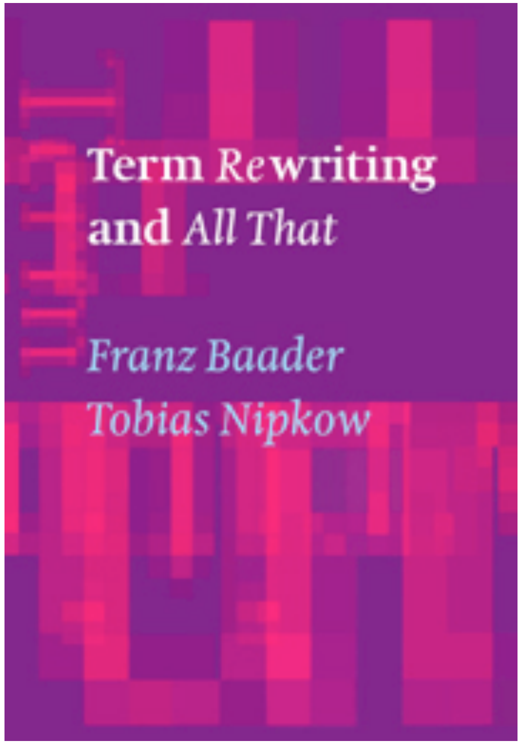
Jacques HERBRAND *(au centre)*
au cours de l'excursion où il trouva la mort

AC and friends

AC-unification (with constants) can be reduced to solving linear diophantine equations, i.e., linear equations in the semi-ring of natural numbers.

[Stickel; 1975]

[Livesey, Siekmann; 1975]



AC and friends

AC-unification (with constants) can be reduced to solving linear diophantine equations, i.e., linear equations in the semiring of natural numbers. [Stickel; 1975] [Livesey, Siekmann; 1975]

Linear constant restrictions can easily be expressed by additional equations.

Can be generalized to the class of commutative/monoidal theories, where unification corresponds to solving linear equations in a corresponding semiring. [Baader; 1989] [Nutt; 1990] [Baader, Nutt; 1996]

Allows us to apply known approaches for solving equations from (computer) algebra (ILP, Gröbner bases, . . .) to decide unifiability and compute unifiers.

Dresden

AC and

AC-unifi
linear dic
ring of n

Linear c

Can be g
where ui
correspc

Allows u
algebra (

Unification in Commutative Theories, Hilbert's Basis Theorem, and Gröbner Bases

FRANZ BAADER

German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

Abstract. Unification in a communitative theory E may be reduced to solving linear equations in the corresponding semiring $S(\mathrm{E})$ [37]. The unification type of E can thus be characterized by algebraic properties of $S(\mathrm{E})$. The theory of Abelian groups with n commuting homomorphisms corresponds to the semiring $\mathbb{Z}[X_1, \ldots, X_n]$. Thus, Hilbert's Basis Theorem can be used to show that this theory is unitary. But this argument does not yield a unification algorithm. Linear equations in $\mathbb{Z}[X_1, \ldots, X_n]$ can be solved with the help of Gröbner Base methods, which thus provide the desired algorithm. The theory of Abelian monoids with a homomorphism is of type zero [4]. This can also be proved by using the fact that the corresponding semiring, namely $\mathbb{N}[X]$, is not Noetherian. Another example of a semiring (even ring) that is not Noetherian is the ring $\mathbb{Z}\langle X_1, \ldots, X_n \rangle$, where $X_1, \ldots, X_n$ $(n > 1)$ are noncommuting indeterminates. This semiring corresponds to the theory of Abelian groups with n noncommuting homomorphisms. Surprisingly, by construction of a Gröbner Base algorithm for right ideals in $\mathbb{Z}\langle X_1, \ldots, X_n \rangle$, it can be shown that this theory is unitary unifying.

Categories and Subject Descriptors: F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*computations on polynomials*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*; *pattern matching*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*mechanical theorem proving*; I.1.2 [**Algebraic Manipulation**]: Algorithms—*algebraic algorithms*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*resolution*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Equational reasoning, Gröbner bases, unification

1. *Introduction*

E-unification is concerned with solving term equations modulo an equational theory E. More formally, let E be an equational theory and $=_{\mathrm{E}}$ be the equality of terms, induced by E. An *E-unification problem* Γ is a finite set of equations $\langle s_i = t_i;\ 1 \le i \le n \rangle_{\mathrm{E}}$ where s_i and t_i are terms. A substitution θ is called an *E-unifier* of Γ iff $s_i\theta =_{\mathrm{E}} t_i\theta$ for each $i, i = 1, \ldots, n$. The set of all E-unifiers of Γ is denoted by $U_{\mathrm{E}}(\Gamma)$.

This research was carried out while the author was a member of IMMD1, University of Erlangen.

Author's address: German Research Center for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany.

Journal of the Association for Computing Machinery, Vol 40, No 3, July 1993, pp 477–503

[Stickel; 1975]

esey, Siekmann; 1975]

onal equations.

ories, [Baader; 1989]

[Nutt; 1990]

[Baader, Nutt; 1996]

rom (computer)
ompute unifiers.

AC and friends

AC-unification (with constants) can be reduced to solving linear diophantine equations, i.e., linear equations in the semiring of natural numbers.

[Stickel; 1975]

[Livesey, Siekmann; 1975]

Linear constant restrictions can easily be expressed by additional equations.

Can be generalized to the class of commutative/monoidal theories, where unification corresponds to solving linear equations in a corresponding semiring.

[Baader; 1989]

[Nutt; 1990]

[Baader, Nutt; 1996]

Allows us to apply known approaches for solving equations from (computer) algebra (ILP, Gröbner bases, . . .) to decide unifiability and compute unifiers.

Commutative/monoidal theories may still have unification type zero.

ACUIh

axiomatizes equivalence in the Description Logic $\mathcal{FL}_0$.

Terminological Cycles in KL-ONE-based Knowledge Representation Languages[1]

Franz Baader

German Research Center for Artificial Intelligence

Projektgruppe WINO, Postfach 2080

D-6750 Kaiserslautern, West Germany

baader@uklirb.uucp

Abstract

Cyclic definitions are often prohibited in terminological knowledge representation languages because, from a theoretical point of view, their semantics is not clear and, from a practical point of view, existing inference algorithms may go astray in the presence of cycles. In this paper, we shall consider terminological cycles in a very small KL-ONE-based language. For this language, the effect of the three types of semantics introduced by (Nebel 1987,1989,1989a) can be completely described with the help of finite automata. These descriptions provide a rather intuitive understanding of terminologies with cyclic definitions and give insight into the essential features of the respective semantics. In addition, one obtains algorithms and complexity results for subsumption determination. As it stands, the greatest fixed-point semantics comes off best. The characterization of this semantics is easy and has an obvious intuitive interpretation. Furthermore, important constructs – such as value-restriction with respect to the transitive or reflexive-transitive closure of a role – can easily be expressed.

language. They proposed to add cyclic definitions which are interpreted by least fixed-point semantics. This was also the starting point for an extensive study of fixed-point extensions of first-order logic (see e.g., (Gurevich & Shelah 1986)).

A thorough investigation of cycles in terminological knowledge representation languages can be found in (Nebel 1987,1989,1989a). Nebel considered three different kinds of semantics – namely, least fixed-point semantics, greatest fixed-point semantics, and what he called descriptive semantics – for cyclic definitions in his language $\mathcal{NTF}$. But, due to the fact that this language is relatively strong[2], he does not provide a deep insight into the meaning of cycles with respect to these three types of semantics. For the two fixed-point semantics, Nebel explicates his point just with a few examples. The meaning of descriptive semantics – which, in Nebel's opinion, comes "closest to the intuitive understanding of terminological cycles" ((Nebel 1989a), p. 124) – is treated more thoroughly. But even in this case the results are not quite satisfactory. For example, the decidability of subsumption

Dresden

Description Logics

AI that is explainable by design

Family of Knowledge Representaion languages of varying expressive power and complexity of reasoning that are tailored towards certain application domains.

P – 2NExpTime

Semantic Web
Mechanical Engineering
Chemical Process Engineering
Biology and Medicine
⋮

Dresden

Description Logics

AI that is explainable by design

Family of Knowledge Representaion languages of varying expressive power and complexity of reasoning that are tailored towards certain application domains.

Decidable fragments of first-order logic often contained in the guarded fragment or the two-variable fragment (with counting).

Explainable by design

- Entailments can in principle be explained using a proof in an appropriated calculus.

  DL entailments often need only a few axioms from the usually large KBs, and mostly have rather small proofs.

  Axiom Pinpointing

- Non-entailments can in principle be explained using a finite counter-model.

  DLs often have the finite model property.

Explainable by design

in principle yes, but …

DL proofs may still be too long or too complicated to be understood by a (non-expert) user.

- How to compute "good" proofs? — CADE 2021 / LPAR 2020
- How to visualize proofs? — DL 2020 and VOILA 2020
- User studies — XLoKR 2020

Generating large DL knowledge bases usually requires considerable manual efforts by knowledge engineers and domain experts.

- Generating DL KBs from finite interpretations — Distel 2011, Borchmann 2014, Kriegel 2019
- Generating medical ontologies from text — J. Biomedical Semantics, 2015
- Repairing DL KBs — CADE 2021, ISWC 2020, JELIA 2019, KR 2018

http://www.galerie-himmel.de/de/Ausstellungen/Kabinett-Gelbke.html

Thanks

- CADE Inc. and the 2020 Herbrand Award Committee chaired by Andrei Voronkov.
- The nominator and the supporters.
- All my co-authors and collaborators.
- The members of my research group past and present.
- DFG and EU for the funding.
- Whomever I forgot to mention.

Spot the winners!