RWTH
LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

On the Complexity of Boolean Unification

Franz Baader

LTCS-Report 97-03

RWTH Aachen
LuFg Theoretische Informatik
http://www-lti.informatik.rwth-aachen.de

Ahornstr. 55
52074 Aachen
Germany

On the Complexity of Boolean Unification

Franz Baader*
LuFg Theoretical Computer Science,
RWTH Aachen
Ahornstraße 55, 52074 Aachen, Germany
e-mail: baader@informatik.rwth-aachen.de

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.

1 Introduction

Boolean unification, i.e., unification modulo the theory of Boolean algebras or rings, has been considered by several authors [5, 14, 13]. 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, 12, 16] 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 [17]. 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, 18].

*Partially supported by the EC Working Group CCL II.

The emphasis in the work on Boolean unification was on developing algorithms that compute a most general unifier for unification problems with constants [5, 14, 13], or finite complete sets of unifiers for general unification problems [17, 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 is only known for elementary unification, where it is easily seen to be NP-complete.

In this research note, we will determine the complexity of the decision problem for more general kinds of Boolean unification problems, namely 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 will show that the decision problem for Boolean unification with constants is Π_2^p-complete whereas the decision problems for Boolean unification with constant restrictions and for general Boolean unification are PSPACE-complete. We will prove these results by establishing a close relationship between the respective unification problems and (certain types of) quantified Boolean formulae [19]. On the one hand, we will make use of a logical characterization [1] of unification problems by certain classes of positive sentences, called conjunctive sentences in the following. On the other hand, we need to show that validity in the theory of Boolean algebras of a special class of such conjunctive sentences, called simple sentences in the following, is equivalent to validity of these simple sentences in the two-element Boolean algebra B_2.

In the next section, we will introduce the relevant definitions from unification theory, and recall the logical characterizations of the different types of unification problems. The third section introduces Boolean unification, shows the connection between Boolean unification and validity of simple sentences, and proves the above mentioned result on the validity of simple sentences in the theory of Boolean algebras. The fourth section puts all these result together, and thus proves the complexity results for Boolean unification.

2 Unification modulo equational theories

An *equational theory* is defined by a set E of identities between terms, i.e., a subset of $T(\Sigma, V) \times T(\Sigma, V)$ for a set of function symbols (signature) Σ and a (countably infinite) set of variables V. With $=_E$ we denote the equational theory defined by E, that is, the least congruence relation on the term algebra $\mathcal{T}(\Sigma, V)$ that is closed under substitutions and contains E. The signature $Sig(E)$ of E is the set of all function symbols occurring in E.

Definition 1 Let E be an equational theory and Δ a signature. An *E-unification problem* over Δ is a finite set of equations

$$P = \{s_1 \stackrel{?}{=}_E t_1, \ldots, s_n \stackrel{?}{=}_E t_n\}$$

between Δ-terms with variables in a (countably infinite) set of variables V. An *E-unifier* of P is a substitution σ such that $s_1\sigma =_E t_1\sigma, \ldots, s_n\sigma =_E t_n\sigma$. The problem P is *E-unifiable* iff it has an E-unifier.

The decision problem for E-unification is the question whether a given E-unification problem over a signature Δ is E-unifiable or not. Depending on the signature Δ, there are three different kinds of instances of the decision problem:

Definition 2 Let E be an equational theory, Δ a signature, and P an E-unification problem over Δ.

- P is an *elementary* E-unification problem iff $\Delta \subseteq \mathit{Sig}(E)$.
- P is an E-unification problem *with constants* iff $\Delta \setminus \mathit{Sig}(E)$ is a set of constant symbols.
- In a *general* E-unification problem P, the set $\Delta \setminus \mathit{Sig}(E)$ may contain arbitrary function symbols.

The constant and function symbols in $\Delta \setminus \mathit{Sig}(E)$ are called *free* constant and function symbols since their interpretation is not constrained by the identities in E. In [1], an additional type of unification problems was introduced, called unification problems with linear constant restrictions:

Definition 3 An E-unification problem *with linear constant restrictions (lcr)* consists of an E-unification problem with constants, P, and a linear ordering $<$ on the variables and free constants occurring in P. A substitution σ is an *E-unifier* of $(P, <)$ iff it is an E-unifier of P that satisfies

$$x < c \quad \text{implies} \quad c \text{ does not occur in } x\sigma$$

for all variables x and free constants c in P.

This kind of unification problems is of interest since a procedure that solves the decision problem for E-unification with lcr can always be turned into a decision procedure for general E-unification. This can be achieved using the nondeterministic combination algorithm described in [1].

The decision problems for elementary unification, unification with constants, and unification with lcr are (polynomial time) equivalent to logical decision problem.

Before we can state these correspondences, we must introduce the relevant classes of sentences. Let E be an equational theory, and $\Sigma := Sig(E)$ be the set of function symbols occurring in E. An atomic Σ-formula is an equation $s = t$. A conjunctive Σ-matrix is a conjunction of atomic Σ-formulae. A *conjunctive Σ-sentence* is a quantifier-prefix followed by a conjunctive Σ-matrix that contains only variables introduced in the prefix. Without loss of generality we assume that the variables occurring in the prefix are all distinct. An *existential conjunctive* Σ-sentence is a conjunctive Σ-sentence whose prefix contains only existential quantifiers, and a *conjunctive AE* Σ-sentence has a prefix consisting of a block of universal quantifiers, followed by a block of existential quantifiers. The conjunctive (existential conjunctive, conjunctive AE) fragment of the equational theory E consists of the set of all conjunctive (conjunctive existential, conjunctive AE) Σ-sentences that are valid in E, i.e., true in all models of E. The decision problem for the conjunctive (conjunctive existential, conjunctive AE) fragment of E is the question whether a given conjunctive (conjunctive existential, conjunctive AE) Σ-sentence belongs to this fragment or not.

Theorem 4 *Let E be an equational theory and $\Sigma := Sig(E)$.*

1. *The decision problems for elementary E-unification and for the conjunctive existential fragment of E can be reduced to each other in linear time.*
2. *The decision problems for E-unification with constants and for the conjunctive AE fragment of E can be reduced to each other in linear time.*
3. *The decision problems for E-unification with lcr and for the conjunctive fragment of E can be reduced to each other in linear time.*
4. *The decision problem for E-unification with lcr can be reduced to the decision problem for general unification in linear time. The nondeterministic polynomial combination algorithm of [1] can be used to reduce the decision problem for general E-unification to the decision problem for E-unification with lcr.*

We just sketch the reductions that can be used to obtain these results. Detailed proofs of the correctness of these reductions can found in [1].
(1) The first statement should be obvious since an elementary E-unification problem can be seen as a conjunction of equational atoms, which is (implicitly) existentially quantified.
(2) By Skolemizing the universally quantified variables, a given conjunctive AE sentence can be turned into a conjunctive existential sentence over a signature enlarged by Skolem constants, and this sentence obviously corresponds to an E-unification problem with constants. Conversely, the free constants of a given E-unification problem with constants can be turned into the universally quantified variables of a corresponding conjunctive AE sentences.

(3) A given E-unification problem with lcr can be turned into a conjunctive sentence as follows: the matrix of this sentence is just the conjunction of the equations in the problem; the variables become existentially quantified variables in the prefix and the free constants universally quantified variables; the order of the quantifiers in the prefix is determined by the linear ordering of the lcr. This reduction can be reversed in the obvious way.
(4) A given E-unification problem with lcr can be turned into the corresponding conjunctive sentence, and from there by Skolemization into an existential sentence over a signature enlarged by Skolem functions, which obviously corresponds to a general E-unification problem. The second statement is an immediate consequence of the fact that the combination algorithm of [1] can be seen as an NP-algorithm which decomposes a given general E-unification problem into a pair consisting of an E-unification problem with lcr and a syntactic unification problem with lcr. Since the syntactic unification problem with lcr can be decided in polynomial time, the remaining problem to be solved is the E-unification problem with lcr. Viewed as a deterministic algorithm, the combination algorithm builds a binary branching search tree of polynomial depth, where at each leaf the above mentioned pair of unification problems must be solved. The original problem is solvable iff there exists a leaf such that both components of its pair are solvable.

3 Boolean unification and validity of simple sentences

The signature Σ_{BA} of Boolean algebras consists of two binary function symbols $+$ and $*$, a unary function symbol $\overline{\ }$, and two constant symbols 0 and 1. The equational theory of Boolean algebras is defined by the following identities:

$$E_{BA} := \left\{ \begin{array}{rclrcl} x+y & = & y+x, & x*y & = & y*x, \\ (x+y)+z & = & x+(y+z), & (x*y)*z & = & x*(y*z), \\ x+(y*z) & = & (x+y)*(x+z), & x*(y+z) & = & (x*y)+(x*z), \\ x+(x*y) & = & x, & x*(x+y) & = & x, \\ x+x & = & x, & x*x & = & x, \\ x+0 & = & x, & x*0 & = & 0, \\ x+1 & = & 1, & x*1 & = & x, \\ x+\overline{x} & = & 1, & x*\overline{x} & = & 0, \\ \overline{x+y} & = & \overline{x}*\overline{y}, & \overline{x*y} & = & \overline{x}+\overline{y}, \\ \overline{\overline{x}} & = & x & & & \end{array} \right\}$$

In many textbooks, one considers $0 \neq 1$ as an additional axiom. We define $T_{BA} := E_{BA} \cup \{0 \neq 1\}$. Obviously, T_{BA} is not a set of identities and thus does not define an equational theory. However, the only difference between T_{BA} and E_{BA} is that the former excludes the trivial one-element model of E_{BA}. The

initial model of E_{BA} is the two-element Boolean algebra B_2, which consists of (the interpretations of) the constants 0 and 1.

Under *Boolean unification* we understand unification modulo E_{BA}. It should be noted that most authors consider the theory of Boolean rings instead of the theory of Boolean algebras. However, since there are linear translations between Boolean ring terms and Boolean algebra terms, this is not a relevant difference.

We may restrict our attention to E_{BA}-unification problems of the form $P := \{s =^{?}_{E_{BA}} 1\}$. This is an obvious consequence of the following simple lemma:

Lemma 5 *Let s and t be terms over a signature containing Σ_{BA}.*

1. $s =_{E_{BA}} t$ *iff* $(s + \overline{t}) * (\overline{s} + t) =_{E_{BA}} 1$.
2. $s =_{E_{BA}} 1$ *and* $t =_{E_{BA}} 1$ *iff* $s * t =_{E_{BA}} 1$.

For the logical characterization of unification problems introduced in the previous section, this means that one can restrict the attention to conjunctive Σ_{BA}-sentences whose matrix consists of a single atomic equation of the form $s = 1$. In the following, we will call such a sentence a *simple Σ_{BA}-sentence*. The main result of this section is the following theorem for simple Σ_{BA}-sentences:

Theorem 6 *Let φ be a simple Σ_{BA}-sentence. Then following statements are equivalent:*

1. *φ is valid in E_{BA}, i.e., it is valid in all models of E_{BA}.*
2. *φ is valid in the initial model B_2 of E_{BA}.*

Proof. $(1 \to 2)$ is trivial since B_2 is a model of E_{BA}. For the proof of $(2 \to 1)$ we use results from model theory, which can, for example, be found in [6]. Assume that φ is valid in B_2.

(a) Since every simple Σ_{BA}-sentence is a Horn sentence (in the sense introduced in [6], p. 407), we can apply Proposition 6.2.2 of [6], which states that validity of Horn sentences is preserved under reduced products. Thus, φ is valid in all reduced products $S(\omega)/D \cong \Pi_D B_2$ of B_2 (see also p. 406 of [6]).

(b) Thus, there does not exist a reduced product $S(\omega)/D$ in which $T_{BA} \cup \{\neg\varphi\}$ holds. Ershov's theorem (Theorem 6.3.20 of [6]) implies that $T_{BA} \cup \{\neg\varphi\}$ is inconsistent, i.e., φ is valid in T_{BA}.

(c) Since $T_{BA} = E_{BA} \cup \{0 \neq 1\}$, this implies that $0 = 1 \vee \varphi$ is valid in E_{BA}. Now assume that B is a model of E_{BA}. If this model satisfies $0 = 1$, then it is of cardinality 1, and thus it trivially satisfies every simple Σ_{BA}-sentence. If it does

not satisfy $0 = 1$, then it must satisfy φ since it satisfies $0 = 1 \vee \varphi$. Consequently, φ is valid in E_{BA}. □

It should be noted that this theorem need not hold for sentences that are not simple. In particular, the argument in (a) does not apply to sentences that are not Horn, and (c) does not apply to sentences that may contain negation.

4 The complexity results

In complexity theory, so-called quantified Boolean formulae have been introduced to obtain a class of problems that is complete for PSPACE [19, 7]. A quantified Boolean formula (QBF) is of the form $(Q_1 x_1) \cdots (Q_n x_n) E$, where E is a Boolean expression involving the propositional variables $x_1, \ldots, x_n$ and $Q_i \in \{\forall, \exists\}$. Validity of such a formula is defined by induction on n: For $n = 0$, the expression E does not contain variables, and it is valid if it evaluates to 1. The formula $(\forall x_1)(Q_2 x_2) \cdots (Q_n x_n) E$ is valid iff both $(Q_2 x_2) \cdots (Q_n x_n) E\{x_1 \mapsto 1\}$ and $(Q_2 x_2) \cdots (Q_n x_n) E\{x_1 \mapsto 0\}$ is valid, and $(\exists x_1)(Q_2 x_2) \cdots (Q_n x_n) E$ is valid iff one of $(Q_2 x_2) \cdots (Q_n x_n) E\{x_1 \mapsto 1\}$ and $(Q_2 x_2) \cdots (Q_n x_n) E\{x_1 \mapsto 0\}$ is valid.

Obviously, a term s over the signature Σ_{BA} can be seen as a Boolean expression E_s. The following lemma is an easy consequence of the definition of validity of a QBF:

Lemma 7 *The simple Σ_{BA}-sentence sentence $(Q_1 x_1) \cdots (Q_n x_n)(s = 1)$ is valid in B_2 iff the corresponding QBF $(Q_1 x_1) \cdots (Q_n x_n) E_s$ is valid.*

An *existential QBF* is a QBF that contains only existential quantifiers, and an *AE QBF* is a QBF whose quantifier prefix consists of a (possibly empty) block of universal quantifiers followed by a (possibly empty) block of existential quantifiers. It is well-known [7] that validity of existential QBFs is NP-complete, validity of AE QBFs is Π_2^p-complete, and validity of QBFs is PSPACE-complete. Thus, Lemma 7, Theorem 6, and Theorem 4 immediately imply:

Theorem 8 *Depending on the kind of unification problems considered, the decision problem for Boolean unification belongs to the following complexity classes:*

1. *Elementary E_{BA}-unification is NP-complete.*
2. *E_{BA}-unification with constants is Π_2^p-complete.*
3. *E_{BA}-unification with lcr and general E_{BA}-unification are PSPACE-complete.*

For the second statement in (3), one should note that the nondeterministic polynomial combination algorithm can easily be realized such that it needs only polynomial space.

5 Conclusion

Decision procedures for unification rather than algorithms computing complete sets of unifiers or most general unifiers are, for example, of interest in constraint-based approaches to theorem proving, term rewriting, and logic programming [4, 15, 10, 9]. In this research note we have determined the exact complexity of the decision problem for Boolean unification. Whereas elementary unification is on the first level of the polynomial hierarchy (NP-complete), unification with constants is on the second level (Π_2^p-complete), and unification with lcr as well as general unification are above the polynomial hierarchy (PSPACE-complete). This is a rather unusual situation since for most of the theories considered until now, the decision problems for unification with constants and unification with lcr are of the same complexity. For example, for *ACI* (which axiomatizes associativity, commutativity, and idempotency of a binary function symbol) and for the theory of Abelian groups, unification with constants and unification with lcr are polynomial, whereas general unification is NP-complete.

As already mentioned in Section 3, Theorem 6, which reduces validity of simple Σ_{BA}-sentences in E_{BA} to validity in B_2, need not hold for more complex sentences. In [11] it is shown that validity of arbitrary Σ_{BA}-sentences is complete for alternating exponential time with a linear number of alternations.[1]

The complexity of computing complete sets of E_{BA}-unifiers for general E_{BA}-unification problems has been investigated by Hermann and Kolaitis [8]. As already mentioned in the introduction, E_{BA} is only finitary for general unification, whereas it is unitary for unification with constants. Hermann and Kolaitis show that even computing the cardinality of a minimal complete set of E_{BA}-unifiers for a given general E_{BA}-unification problem is a $\#P$-hard, which implies that this function cannot be computed in polynomial time, unless P = NP.

Acknowledgment: I should like to thank Miki Hermann for alerting me to the fact that the decision procedures for Boolean unification with constants found in the literature are in Π_2^p and not in NP.

References

[1] F. Baader and K.U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symbolic Computation*, 21:211–243, 1996.

[2] G. Boole. *The Mathematical Analysis of Logic: Being an Essay towards a Calculus of Deductive Reasoning.* Macmillan, Barclay, and Macmillan, Cambridge, 1847.

[1] This is a complexity class that lies even above nondeterministic exponential time.

[3] A. Boudet, J.-P. Jouannaud, and M. Schmidt-Schauß. Unification in Boolean rings and Abelian groups. *J. Symbolic Computation*, 8:449–477, 1989.

[4] H.-J. Bürckert. *A Resolution Principle for a Logic with Restricted Quantifiers*, volume 568 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1991.

[5] W. Büttner and H. Simonis. Embedding Boolean expressions into logic programming. *J. Symbolic Computation*, 4(2):191–205, 1987.

[6] C.C. Chang and H.J. Keisler. *Model Theory, Third Edition*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, The Netherlands, 1990.

[7] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-completeness*. Freeman, New York, 1979.

[8] M. Hermann and P.G. Kolaitis. Unification algorithms cannot be combined in polynomial time. In M.A. McRobbie and J.K. Slaney, editors, *Proceedings of the 13th International Conference on Automated Deduction*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 246–260. Springer-Verlag, 1996.

[9] J. Jaffar and J.-L. Lassez. Constraint logic programming. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, pages 111–119, Munich, Germany, 1987.

[10] C. Kirchner and H. Kirchner. Constrained equational reasoning. In *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, Portland, Oregon, 1989. ACM Press.

[11] D. Kozen. Complexity of Boolean algebras. *Theoretical Computer Science*, 10(3):221–247, 1980.

[12] L. Löwenheim. Über das Auflösungsproblem im logischen Klassenkalkül. *Sitzungsberichte Berliner Math. Gesell.*, 7:89–94, 1908.

[13] U. Martin and T. Nipkow. Boolean unification—The story so far. *J. Symbolic Computation*, 7(3,4):275–293, 1989.

[14] U. Martin and T. Nipkow. Unification in Boolean rings. *J. Automated Reasoning*, 4:381–396, 1989.

[15] R. Nieuwenhuis and A. Rubio. AC-supperposition with constraints: No AC-unifiers needed. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 545–559, Nancy, France, 1990. Springer-Verlag.

[16] Sergiu Rudeanu. *Boolean Functions and Equations*. North-Holland, 1974.

[17] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symbolic Computation*, 8(1,2):51–99, 1989.

[18] H. Simonis and M. Dincbas. Using an Extended Prolog for Digital Circuit Design. In *IEEE International Workshop on AI Applications to CAD Systems for Electronics*, pages 165–188, Munich, Germany, 1987.

[19] L.J. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. In *Proc. 5th ACM Symp. on Theory of Computing*, pages 1–9, New York, 1973. Association for Computing Machinery.