On Deciding Subsumption Problems* (extended abstract)

Uwe Egly Reinhard Pichler Stefan Woltran†
Technische Universität Wien

Abstract

Subsumption is an important redundancy elimination method in automated theorem proving. For a given Herbrand universe H, it can be strengthened to the *clausal H-subsumption*, i.e., a clause D over a finite or infinite universe H is H-subsumed by a clause set $\mathcal{C}$, iff, for every H-ground instance $D\vartheta$ of D, there is a clause $C \in \mathcal{C}$, such that C subsumes $D\vartheta$. If the clauses in $\mathcal{C}$ and the clause D are atoms, then we speak about *atomic H-subsumption*.

Clausal and atomic H-subsumption have gained increasing importance in the field of automated model building as recent publications in this field demonstrate. Moreover, problems equivalent to atomic H-subsumption have been studied with different terminologies in many areas of Computer Science.

Both clausal and atomic H-subsumption are known to be intractable, i.e., Π_2^p-complete and NP-complete, respectively. In this paper, we present a new approach to deciding (clausal and atomic) H-subsumption that is based on a reduction to QSAT and SAT, respectively.

1 Introduction

Redundancy elimination is one of the most important tasks in all kinds of deduction systems. One important redundancy elimination technique is *subsumption* which is implemented in all practically successful saturation-based theorem provers. Although subsumption is a computationally expensive operation (see problem LO18 in [9] for the NP-completeness result), omitting it from implementations would be even worse because redundant clauses spoil the search space, preventing the theorem prover to find a proof.

One goal of this paper is the development of redundancy elimination techniques even stronger than subsumption, which are implemented as a reduction to SAT and restricted forms of QSAT (satisfiability of *quantified* Boolean formulas). The two techniques are *atomic H-subsumption* and *clausal H-subsumption* which have various applications in automated model building [8, 4, 3], machine learning [16], logic programming [10, 15], functional programming [15], and as an extension of many familiar resolution- and paramodulation-based inference systems without destroying the refutational completeness [20]. In what follows, we briefly recall the definitions of the important problems:

*The work was partially supported by the Austrian Science Foundation under grant P15068.

†Corresponding author. Abteilung für Wissensbasierte Systeme E184/3, TU Wien, Favoritenstr. 9–11, A-1040 Wien, AUSTRIA, Phone: +43-1-58801-18466, stefan@kr.tuwien.ac.at

SUBSUMPTION (first-order subsumption)
Input: A clause set $\mathcal{C} \equiv \{C_1, \ldots, C_n\}$ and a clause D with $C_i \equiv L_{i1} \vee \ldots \vee L_{il_i}$ and $D \equiv M_1 \vee \ldots \vee M_m$.
Question: Does $\mathcal{C}$ subsume D, i.e., does there exist a clause $C_i \in \mathcal{C}$ and a substitution ϑ on the variables occurring in C_i, such that $\{L_{i1}\vartheta, \ldots, L_{il_i}\vartheta\} \subseteq \{M_1, \ldots, M_m\}$? If this is the case, then we write $\mathcal{C} \leq_{ss} D$.
ATOM-HSUB (atomic H-subsumption) over a given Herbrand universe H.
Input: A set of atoms $\mathcal{A} \equiv \{A_1, \ldots, A_n\}$, a single atom B, and H.
Question: Is every H-ground instance[1] $B\vartheta$ of B an instance of some atom A_i? If this is the case, then we write $\mathcal{A} \leq_{ss}^{H} B$.
CLAUSE-HSUB (clausal H-subsumption) over a given Herbrand universe H.
Input: A clause set $\mathcal{C} \equiv \{C_1, \ldots, C_n\}$, a single clause D, and H.
Question: Is every H-ground instance $D\vartheta$ of D subsumed by some clause C_i? If this is the case, then we write $\mathcal{C} \leq_{ss}^{H} D$.

By the definition of subsumption, $\mathcal{C} \leq_{ss} D$ implies $\mathcal{C} \leq_{ss} D\vartheta$ for any substitution ϑ over any Herbrand universe H. Hence, (ordinary) subsumption implies H-subsumption for any H. But the converse is, in general, not true as the following example illustrates. Let $\mathcal{C} \equiv \{P(x,y) \vee Q(x) \vee Q(y)\}$ and $D \equiv P(x,y) \vee Q(a) \vee Q(b)$. Then $\mathcal{C} \not\leq_{ss} D$, since there is no substitution ϑ with domain $\{x, y\}$, such that, on the one hand, $P(x,y)\vartheta \in D$ and, on the other hand, $Q(x)\vartheta \in D$ and $Q(y)\vartheta \in D$. But for $H = \{a, b\}$, $\mathcal{C} \leq_{ss}^{H} D$. Let $D\vartheta$ be an H-ground instance of D, i.e., $x\vartheta \in \{a, b\}$ and $y\vartheta \in \{a, b\}$. Then $\mathcal{C}\vartheta \leq_{ss} D\vartheta$. Contrary, for $H = \{a, b, c\}$, $\mathcal{C} \not\leq_{ss}^{H} D$. Consider the H-ground substitution $\vartheta = \{x \leftarrow c, y \leftarrow c\}$. Then the H-ground instance $D\vartheta$ of D is not subsumed by $\mathcal{C}$.

Finally, let us consider the computational complexity results for the H-subsumption problems. Clausal H-subsumption is Π_2^p-complete (cf. [18] for the Π_2^p-hardness proof and [11] for the Π_2^p-membership proof). The coNP-completeness of ATOM-HSUB has been shown independently by several authors (cf. [14], [12], [13]).

For all of the three decision problems mentioned above, efficient algorithms are an important issue. In this paper, we focus on H-subsumption and contribute to the search for efficient algorithms by transforming H-subsumption problems in such a way that they can be solved by theorem provers for (quantified) propositional logic, even if the Herbrand universe H is *infinite*. By considering the complementary problems, we shall thus reduce the complementary problem co-ATOM-HSUB to SAT and the co-CLAUSE-HSUB problem to QSAT_2, i.e., to the evaluation problem for quantified Boolean formulas with one quantifier alternation. From a theoretical point of view, these *polynomial* translations are optimal because they reduce an NP-complete problem to SAT and a problem complete for Σ_2^p (which is the complementary class to Π_2^p) to QSAT_2 which is itself Σ_2^p-complete [21].

In this work, *equational problems* constitute the crucial link between H-subsumption problems and (quantified) satisfiability. First of all, subsumption problems are naturally expressible by means of equational problems. Secondly, we construct a polynomial mapping from equational problems over a finite domain into the language of quantified Boolean formulas. Finally, we show how equational problems representing an H-subsumption problem over an infinite H, can polynomially be transformed into equivalent equational problems over some finite universe. So, the latter problems then are once again reducible to (quantified) satisfiability.

[1] An H-ground instance results by replacing variables by elements from the given Herbrand universe H.

2 Equational Problems

Equational problems are first-order formulas, whose only predicate symbol is syntactic equality "=". Hence, atoms of equational problems are either of the form $s = t$ for some terms s and t or the trivial problems $\top$ and $\bot$. $\top$ is the trivially true problem and $\bot$ is the trivially false problem, respectively. Disequations $s \neq t$ are used as a short-hand notation for negated equations $\neg(s = t)$.

An equational problem $\mathcal{P}$ is usually considered over some fixed finite or infinite Herbrand universe H or, equivalently, over some fixed finite signature Σ consisting of function symbols of arity ≥ 0. An interpretation of $\mathcal{P}$ is given through an H-ground substitution σ, whose domain coincides with the set of free variables of the equational problem and whose image is (a subset of) the Herbrand universe H. The trivial problem $\top$ evaluates to "true" in every interpretation. Likewise, $\bot$ always evaluates to "false". A single equation $s = t$ is validated by an H-ground substitution σ, if and only if $s\sigma$ and $t\sigma$ are syntactically identical. The connectives $\wedge$, $\vee$, $\neg$, and $\rightarrow$, as well as the quantifiers $(\exists x)$, $(\forall x)$ are interpreted "as usual". An H-ground substitution σ which validates a problem $\mathcal{P}$ is called a *solution* of $\mathcal{P}$. An equational problem is *satisfiable* over the Herbrand universe H (or, for short, H-satisfiable), if and only if it has at least one solution.

In what follows, we give a natural translation of the decision problems under consideration into equational problems. We start with some ancillary notation. Assume that we have, for any predicate symbol P of arity n, new function symbols f_P^+, f_P^- of the same arity as P. Then, L^* denotes $f_P^+(t_1, \ldots, t_n)$ if $L \equiv P(t_1, \ldots, t_n)$ and $f_P^-(t_1, \ldots, t_n)$ if $L \equiv \neg P(t_1, \ldots, t_n)$, respectively.

Let H be an arbitrary Herbrand universe. Moreover, let $A_1, \ldots, A_n, B$ be atoms over H and $\vec{x}$ and $\vec{y}$ denote vectors of variables with $Var(A_i) \subseteq \vec{y}$ for every i, $Var(B) \subseteq \vec{x}$ and $\vec{x} \cap \vec{y} = \emptyset$, where $Var(F)$ denotes the set of variables occurring in a formula F. Then $\{A_1, \ldots, A_n\} \not\leq_{ss}^H B$ holds, iff the following equational problem is H-satisfiable:

$$(\exists \vec{x})(\forall \vec{y}) \bigwedge_{i=1}^{n} (A_i^* \neq B^*) \tag{1}$$

The correctness of this problem reduction is immediate from the definition of atomic H-subsumption, namely that $\{A_1, \ldots, A_n\} \not\leq_{ss}^H B$ holds, iff there exists an H-ground substitution ϑ for the variables $\vec{x}$, such that $B\vartheta$ is not an instance of any A_i, i.e., for all possible variable substitutions σ (over H) of the variables $\vec{y}$ and for all $i \in \{1, \ldots, n\}$, the disequation $A_i^*\sigma \neq B^*\vartheta$ holds. Actually, a similar reduction to equational problems can be found in [4].

For co-CLAUSE-HSUB we generalize the above idea. Let $\mathcal{C} \equiv \{C_1, \ldots, C_n\}$ be a clause set with $C_i \equiv L_{i1} \vee \ldots \vee L_{il_i}$ and D a clause with $D \equiv M_1 \vee \ldots \vee M_m$. Moreover, let all L_{ij} and M_k be literals over H and let $\vec{x}$ and $\vec{y}$ denote vectors of variables with $Var(\mathcal{C}) \subseteq \vec{y}$, $Var(D) \subseteq \vec{x}$ and $\vec{x} \cap \vec{y} = \emptyset$. Then, $\mathcal{C} \not\leq_{ss}^H D$ holds, iff the following equational problem

$$(\exists \vec{x})(\forall \vec{y}) \bigwedge_{i=1}^{n} \bigvee_{j=1}^{l_i} \bigwedge_{k=1}^{m} (L_{ij}^* \neq M_k^*) \tag{2}$$

is satisfiable over H. Clearly, systems of form (2) coincide with (1) iff $l_i = 1$ for all $i = 1 \ldots n$, $m = 1$, and all L_{ij} and M_k are positive literals, i.e., iff an atomic H-subsumption problem is given.

3 H-Subsumption over a Finite Universe

This section outlines how a certain class of *closed* equational problems over some fixed finite Herbrand universe H can be expressed in terms of (quantified) propositional logic. More formally, this class, denoted by $\mathcal{C}_H$, restricts atoms of equational problems to be of the form $f(x_1, \ldots, x_{n(f)}) = g(y_1, \ldots, y_{n(g)})$ where each argument x_i, y_i is either a first-order variable or some element from H. Clearly, this class contains all problems of the forms (1) and (2), respectively, over some finite H.

We consider the following underlying alphabet of propositional atoms At_H, containing for each first-order variable x a set of mutually disjoint propositional atoms $x_1, \ldots, x_N$ with N given by $\lceil \mathrm{ld}(card(H)) \rceil$. Moreover, elements a in H are represented by a binary encoding using mutually disjoint vectors of propositional constants $\vec{a} \equiv \langle a_1, \ldots, a_N \rangle$ with $a_i \in \{\top, \bot\}$. The basic idea is to express $x = y$ in the language of propositional logic by a conjunction of equivalences

$$x \Leftrightarrow_N y \equiv (x_1 \leftrightarrow y_1) \wedge \ldots \wedge (x_N \leftrightarrow y_N) \tag{3}$$

where x, y are either first-order variables or elements from H and $p \leftrightarrow q$ represents $(p \rightarrow q) \wedge (q \rightarrow p)$. Here, we have to be careful, whether assignments to propositional atoms from At_H have corresponding first-order assignments. For instance, consider H is given by $\{a, b, c\}$ and the corresponding vectors are $\vec{a} \equiv \langle \bot, \bot \rangle$, $\vec{b} \equiv \langle \bot, \top \rangle$, and $\vec{c} \equiv \langle \top, \bot \rangle$. Indeed, we should exclude assignments for x_i corresponding to $\langle \top, \top \rangle$. In other words, we have to define some form of *domain closure axiom* (DCA) which restricts truth-assignments to propositional atoms $x_1, \ldots, x_n$ in such a way that they represent only elements from H. Straightforwardly, we define the propositional module

$$DCA[x] \equiv \bigvee_{a \in H} (x \Leftrightarrow_N a), \tag{4}$$

which is used frequently later on. In certain cases, $DCA[x]$ can be optimized. For instance, if $card(H) = 2^N$ then $DCA[x]$ reduces to $\top$ for any variable x.

In order to obtain a *polynomial* mapping from equational problems into a propositional language we have to use a well-known generalization of propositional logic, namely quantified Boolean formulas (QBFs), which additionally allow for quantifications over propositional atoms. These quantifications are—similar to first-order logic—either existential or universal. To avoid ambiguities we use the symbols $\tilde{\exists}$, $\tilde{\forall}$. Informally, a QBF of the form $\tilde{\forall} p \, \tilde{\exists} q \, \Phi$ means that for all truth assignments of p there is a truth assignment of q such that Φ is true.

The construction of the overall reduction is as follows: Let $\mathcal{P} \in \mathcal{C}_H$ and $N = \lceil \mathrm{ld}(card(H)) \rceil$. Then, $\mathcal{T}[\mathcal{P}]$ is recursively obtained using abbreviations (3) and (4):

1. $\mathcal{T}[\top] \equiv \top$; $\mathcal{T}[\bot] \equiv \bot$.
2. For atoms $s = t$ with $s \equiv f(x_1, \ldots, x_{n(f)})$ and $t \equiv g(y_1, \ldots, y_{n(g)})$, we define

$$\mathcal{T}[s = t] \equiv \bot \text{ if } f \neq g$$

and otherwise (we have $n(f) = n(g)$)

$$\mathcal{T}[s = t] \equiv \bigwedge_{i=1}^{n(f)} \big(x_i \Leftrightarrow_N y_i\big).$$

3. For an equational problem $\mathcal{P} \equiv \neg\mathcal{P}_1$, we define $\mathcal{T}[\mathcal{P}] \equiv \neg\mathcal{T}[\mathcal{P}_1]$.

4. For equational problems $\mathcal{P}_1$, $\mathcal{P}_2$ and $\circ \in \{\wedge, \vee, \rightarrow\}$, we define

$$\mathcal{T}[\mathcal{P}_1 \circ \mathcal{P}_2] \equiv \mathcal{T}[\mathcal{P}_1] \circ \mathcal{T}[\mathcal{P}_2].$$

5. For an equational problem $\mathcal{P}$, $\mathsf{Q} \in \{\exists, \forall\}$, and x a first-order variable, we have

$$\mathcal{T}[\mathsf{Q}x\,\mathcal{P}] \equiv \tilde{\mathsf{Q}}x_1 \ldots \tilde{\mathsf{Q}}x_N \Big(DCA[x] \circ_{\mathsf{Q}} \mathcal{T}[\mathcal{P}]\Big),$$

where $\circ_\exists = \wedge$, and $\circ_\forall = \rightarrow$.

We claim that $\mathcal{P} \in \mathcal{C}_H$ is satisfiable over H iff the QBF $\mathcal{T}[\mathcal{P}]$ evaluates to true. But then, we immediately get a procedure to decide problems co-ATOM-HSUB and co-CLAUSE-HSUB by means of QBFs, whenever H is finite. We outline the more general case, which maps the problem co-CLAUSE-HSUB into the evaluation problem for QBFs with one quantifier alternation.

Theorem 1 *Let H be a* finite *Herbrand universe, $\mathcal{C} \equiv \{C_1, \ldots, C_n\}$ a clause set with $C_i \equiv L_{i1} \vee \ldots \vee L_{il_i}$ and D a clause with $D \equiv M_1 \vee \ldots \vee M_m$, where all L_{ij} and M_k are literals over H. Let $\vec{x}$ and $\vec{y}$ denote vectors of variables with $Var(\mathcal{C}) \subseteq \vec{y}$, $Var(D) \subseteq \vec{x}$ and $\vec{x} \cap \vec{y} = \emptyset$.*

Then, $\mathcal{C} \not\leq_{ss}^{H} D$ holds iff the QBF $\mathcal{T}[\mathcal{P}_{\mathcal{C},D}]$ evaluates to true where $\mathcal{P}_{\mathcal{C},D}$ is of the form

$$(\exists\vec{x}\,)(\forall\vec{y}\,) \bigwedge_{i=1}^{n} \bigvee_{j=1}^{l_i} \bigwedge_{k=1}^{m} (L_{ij}^* \neq M_k^*).$$

Obviously, $\mathcal{T}[\mathcal{P}_{\mathcal{C},D}]$ can be constructed in polynomial time. Moreover, one can show that, for the restricted case of atomic H-subsumption, the universal quantification can be omitted by reducing $\mathcal{T}[\mathcal{P}_{\mathcal{C},D}]$ to an equivalent SAT-problem without exponential increase of formula length. Thus, both problems co-ATOM-HSUB and co-CLAUSE-HSUB are reduced to evaluation problems reflecting the inherent complexity of the original tasks.

4 H-Subsumption over an Infinite Universe

4.1 Atomic H-Subsumption over an Infinite Universe

Let $\mathcal{P} \equiv (\exists\vec{x}\,)(\forall\vec{y}\,) \bigwedge_{i=1}^{n}(A_i^* \neq B^*)$ be an equational problem over the infinite universe H obtained from a co-ATOM-HSUB problem by the reduction described in Section 2. Then we can construct in *polynomial time* an equivalent equational problem $\mathcal{P}'$ over a finite universe H'. This construction consists of two steps:

1. First we transform the equational problem $\mathcal{P}$ over H into an equivalent equational problem $\mathcal{Q}$ over the same universe H but with a much simpler form. In particular, we replace the original quantifier prefix $\exists^* \forall^*$ by a purely existential quantifier prefix.

2. From the resulting problem $\mathcal{Q}$ over H it is then possible to define an equational problem $\mathcal{P}'$ over a finite universe H', such that $\mathcal{Q}$ (and, equivalently $\mathcal{P}$) is H-satisfiable iff $\mathcal{P}'$ is H'-satisfiable.

The details and a full correctness proof of this problem transformation can be found in [7]. In the remainder of this section, we illustrate the main ideas of this transformation by means of a simple example. Let $\mathcal{P} \equiv (\exists \vec{x})(\forall \vec{y})\,(A^* \neq B^*)$ be an equational problem over H where $A \equiv P(f(f(y_3, y_1), y_1), f(y_1, y_2))$, $B \equiv P(f(x_1, g(x_2), f(x_3, g(a)))$, and with signature $\Sigma = \{a, f, g\}$, where a is a constant and the arities of f and g are given by 2 and 1, respectively.

Actually, $A^* \neq B^*$ is equivalent to $x_1 \neq f(y_3, g(x_2)) \vee x_3 \neq g(x_2) \vee y_2 \neq g(a) \vee y_1 \neq g(x_2)$, by *unification*.[2] Moreover, it can be shown that all disequations of the form $y \neq t$, where y is a universally quantified variable, may be deleted (cf. the U_2*-rule* from [6]). Hence, we are left with the disjunction $x_1 \neq f(y_3, g(x_2)) \vee x_3 \neq g(x_2)$.

Our goal is to transform the disequations into the simple form $x_i \neq x_j$, where x_i and x_j are two distinct, existentially quantified variables. To this end, we make use of the *explosion rule* from [6] as follows. Observe that the disjunction $(\exists \vec{z})[t = a \vee t = g(z_1) \vee t = f(z_2, z_3)]$ holds over our signature $\Sigma = \{a, f, g\}$ for any term $t \in H$. In particular, we may apply the explosion rule to the variable x_1. Hence, $\mathcal{P}$ is equivalent to

$$(\exists \vec{x})(\forall \vec{y})[x_1 = a \vee x_1 = g(x_4) \vee x_1 = f(x_5, x_6)] \wedge [x_1 \neq f(y_3, g(x_2)) \vee x_3 \neq g(x_2)]. \quad (5)$$

The equation $x_1 = a$ and also $x_1 = g(x_4)$ clearly imply the disequation $x_1 \neq f(y_3, g(x_2))$ and hence the whole disjunction of the second conjunct of (5). We thus further transform the problem (5) into

$$(\exists \vec{x})(\forall \vec{y})\Big(x_1 = a \vee x_1 = g(x_4) \vee [x_1 = f(x_5, x_6) \wedge (x_1 \neq f(y_3, g(x_2)) \vee x_3 \neq g(x_2))]\Big).$$

Due to the conjunction with $x_1 = f(x_5, x_6)$, we may transform $x_1 \neq f(y_3, g(x_2))$ into $f(x_5, x_6) \neq f(y_3, g(x_2))$, which in turn is equivalent to $x_5 \neq y_3 \vee x_6 \neq g(x_2)$. The disequation $x_5 \neq y_3$ has an isolated universally quantified variable as an argument and may therefore be deleted (again by the U_2-rule from [6]). Moreover, the disjunction $x_6 \neq g(x_2) \vee x_3 \neq g(x_2)$ may be transformed into $x_6 \neq x_3 \vee x_3 \neq g(x_2)$. This is basically due to the equivalence $\neg F \vee \neg G \equiv (\neg F \wedge G) \vee \neg G$, which holds for any logical formulas F, G. It should be noted that this replacement of $g(x_2)$ by x_3 in the first disequation is crucial for the polynomial upper bound on the time complexity, since it guarantees that the number of applications of the explosion rule is bounded by the number of nodes in the dag (directed acyclic graph) representation of $A^* \neq B^*$.

By repeating the above steps we can also simplify the disequation $x_3 \neq g(x_2)$. In total, we get $\mathcal{Q}$ as

$$\begin{aligned}(\exists \vec{x})\Big(& x_1 = a \vee x_1 = g(x_4) \vee (x_1 = f(x_5, x_6) \wedge x_3 = a) \vee (x_1 = f(x_5, x_6) \wedge x_3 = f(x_8, x_9)) \vee \\ & [(x_1 = f(x_5, x_6) \wedge x_3 = g(x_7)) \wedge (x_6 \neq x_3 \vee x_7 \neq x_2)]\Big).\end{aligned}$$

From this we define the desired equational problem $\mathcal{P}'$ over the finite universe H', as follows. We set $H' \equiv \{b_1, \ldots, b_3, c_1, \ldots, c_7\}$, i.e., H' contains one constant b_i for each symbol in Σ and one constant c_i for each variable in $\mathcal{Q}$. Moreover, $\mathcal{P}'$ is defined as

$$\begin{aligned}\mathcal{P}' \equiv (\exists \vec{x})\Big(& x_1 = b_1 \vee x_1 = b_3 \vee (x_1 = b_2 \wedge x_3 = b_1) \vee (x_1 = b_2 \wedge x_3 = b_2) \vee \\ & [(x_1 = b_2 \wedge x_3 = b_3) \wedge (x_6 \neq x_3 \vee x_7 \neq x_2)]\Big).\end{aligned}$$

[2]Of course, unifiers have to be represented as dags (directed acyclic graphs) in order to avoid an exponential blow-up.

In other words, the constants b_1, b_2, and b_3 were introduced in order to encode terms of the form a, $f(x_i, x_j)$ and $g(x_k)$, which we already know from the explosion rule. The idea of the constants c_i is to guarantee that any conjunction of disequations over H' is satisfiable, provided that none of the disequations is of the trivially false form $t \neq t$. Of course, for this simple example, we could have chosen a much smaller universe H'. However, for the general case treated in [7], this is indeed needed.

4.2 Clausal H-Subsumption over an Infinite Universe

It turns out that the reduction of the co-CLAUSE-HSUB problem into an equational problem according to Section 2 is not appropriate in case of an infinite universe. Instead, we need a different approach, which will be briefly sketched below. Actually, we shall only show how an H-subsumption problem of the form $C \not\leq_{ss}^{H} D$ with clauses $C \equiv L_1 \vee \ldots \vee L_l$ and $D \equiv M_1 \vee \ldots \vee M_m$ can be reduced to an equational problem over a finite universe. Note that this special case of H-subsumption by a single clause C is just as hard as H-subsumption by a a clause set $\mathcal{C}$, i.e., they are both Π_2^p-hard (cf. [18]). The general case, $\mathcal{C} \not\leq_{ss}^{H} D$, as well as any details and proofs are left to [7].

Before we start with our problem reduction, we introduce some useful notation: We write $\langle L_{i_1}, \ldots, L_{i_k} \rangle$ and $\langle M_{j_1}, \ldots, M_{j_k} \rangle$, respectively, in order to denote tuples of literals. Moreover, we write $\langle L_{i_1}, \ldots, L_{i_k} \rangle \leq_s \langle M_{j_1}, \ldots, M_{j_k} \rangle$ to denote that $\langle M_{j_1}, \ldots, M_{j_k} \rangle$ is an instance of $\langle L_{i_1}, \ldots, L_{i_k} \rangle$, i.e., there exists a substitution σ, such that $L_{i_\alpha}\sigma = M_{j_\alpha}$ for every α. Then our problem reduction goes as follows:

First of all, we recall the following criterion for $C \not\leq_{ss} D$ (i.e., for ordinary subsumption): For every tuple $(j_1, \ldots, j_l) \in \{1, \ldots, m\}^l$, $\langle L_1, \ldots, L_l \rangle \not\leq_s \langle M_{j_1}, \ldots, M_{j_l} \rangle$ holds. Likewise, $C \not\leq_{ss}^{H} D$ is equivalent to the condition that there exists an H-ground substitution ϑ on the variables in $Var(D)$, such that for every tuple $(j_1, \ldots, j_l) \in \{1, \ldots, m\}^l$, $\langle L_1, \ldots, L_l \rangle \not\leq_s \langle M_{j_1}\vartheta, \ldots, M_{j_l}\vartheta \rangle$ holds.

Recall how a straightforward matching algorithm detects the above condition: It either finds a non-variable position p in $\langle L_1, \ldots, L_l \rangle$, such that the leading symbol of the subterm of $\langle L_1, \ldots, L_l \rangle$ at position p is different from the leading symbol of the corresponding subterm at position p in $\langle M_{j_1}\vartheta, \ldots, M_{j_l}\vartheta \rangle$. Or it finds two variable positions p_1 and p_2 in $\langle L_1, \ldots, L_l \rangle$ with the same variable y, such that these two occurrences of y in $\langle L_1, \ldots, L_l \rangle$ are matched against two distinct terms at the corresponding positions in $\langle M_{j_1}\vartheta, \ldots, M_{j_l}\vartheta \rangle$. In other words, it suffices to consider pairs of literals rather than the whole l-tuples $\langle L_1, \ldots, L_l \rangle$ and $\langle M_{j_1}\vartheta, \ldots, M_{j_l}\vartheta \rangle$. We thus get the following condition: $C \not\leq_{ss}^{H} D$, iff there exists an H-ground substitution ϑ on the variables in $Var(D)$, such that for every tuple $(j_1, \ldots, j_l) \in \{1, \ldots, m\}^l$, there exist two indices $\alpha < \beta \in \{1, \ldots, l\}$ with $\langle L_\alpha, L_\beta \rangle \not\leq_s \langle M_{j_\alpha}\vartheta, M_{j_\beta}\vartheta \rangle$.

Now let $(\alpha, \beta, \gamma, \delta)$ be an arbitrary quadruple of indices with $\langle L_\alpha, L_\beta \rangle \not\leq_s \langle M_\gamma\vartheta, M_\delta\vartheta \rangle$. Then the following equational problem over the infinite Herbrand universe H is satisfiable:

$$(\exists \vec{x}\,)(\forall \vec{y}\,)\langle L_\alpha^*, L_\beta^* \rangle \neq \langle M_\gamma^*, M_\delta^* \rangle$$

By the considerations in Section 4.1, we know how to transform such an equational problem $\mathcal{P}$ into a purely existentially quantified equational problem $\mathcal{P}'$ over the *finite* Herbrand universe H'. For a given quadruple $(\alpha, \beta, \gamma, \delta)$, we shall write $(\exists \vec{x}\,)P(\alpha, \beta, \gamma, \delta)$ to denote this equational problem $\mathcal{P}'$. Hence, we can test whether $C \not\leq_{ss}^{H} D$ holds, by checking whether the following "equational problem"

$$(\exists \vec{x}\,)(\forall j_1, \ldots, j_l)(\exists \alpha, \beta)(\alpha < \beta) \wedge P(\alpha, \beta, j_\alpha, j_\beta)$$

is satisfiable. Actually, we have slightly cheated in the formulation of the above equational problem, since each problem $P(\alpha, \beta, j_\alpha, j_\beta)$ is somehow parameterized by the variables α, β, j_α, and j_β. Moreover, we would like to get rid of the quantification $(\exists \alpha, \beta)$ (and also the inequality $\alpha < \beta$), in order to arrive at the desired quantifier prefix $\exists^* \forall^*$. But this can be easily achieved in the following way.

1. We replace $(\exists \alpha, \beta)(\alpha < \beta) \wedge P(\alpha, \beta, j_\alpha, j_\beta)$ by the following disjunction of $\frac{l \times (l-1)}{2}$ disjuncts $P(1, 2, j_1, j_2) \vee \ldots \vee P(1, l, j_1, j_l) \vee P(2, 3, j_2, j_3) \vee \ldots \vee P(l-1, l, j_{l-1}, j_l)$.

2. Each problem $P(\alpha, \beta, j_\alpha, j_\beta)$ is replaced by a conjunction of m^2 conjuncts of the form $[(j_\alpha = 1 \wedge j_\beta = 1) \rightarrow P(\alpha, \beta, 1, 1)] \wedge \ldots \wedge [(j_\alpha = m \wedge j_\beta = m) \rightarrow P(\alpha, \beta, m, m)]$.

In total, we get the equational problem $(\exists \vec{x}\,)(\forall j_1, \ldots, j_l)\mathcal{Q}$ with two sorts of variables, namely $\vec{x}$ over the finite Herbrand universe H' and $j_1, \ldots, j_l$ over the finite universe $\{1, \ldots, m\}$. Moreover, $\mathcal{Q}$ is a disjunction of $\frac{l \times (l-1)}{2}$ disjuncts, each of which is a conjunction of m^2 conjuncts.

Note that the reduction of equational problems over a finite universe in Section 3 can be easily extended to deal with equational problems containing variables of different sorts (provided of course that all of these sorts are finite!). Hence, for both problems, i.e., clausal and atomic H-subsumption over an infinite Herbrand universe H we have constructed a reduction to equivalent equational problems over finite Herbrand universes, which according to Section 3 are solvable by means of QBFs.

5 Concluding Remarks

In this paper, we have presented a new approach to deciding (clausal and atomic) H-subsumption via a reduction to the well studied QSAT- and SAT-problem, respectively. In particular, clausal H-subsumption has so far suffered from the lack of implementations, which is a genuine obstacle to its wider use. With the new method presented here, all the progress that has been made in the construction of efficient SAT- and QBF-solvers over the past years, is now also accessible to the H-subsumption problem. This is a first step towards more efficient H-subsumption algorithms, to which further steps should be added. It will then be interesting, to actually integrate clausal H-subsumption into an automated theorem prover. Practical experimentation will show in which cases the additional cost of H-subsumption (compared with ordinary subsumption) is outweighed by the gained power of redundancy elimination. As a byproduct, we have shown how equational problems representing an H-subsumption problem over an infinite H, can polynomially be transformed into equivalent equational problems over some finite universe.

References

[1] F. Baader and J. H. Siekmann. Unification theory. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, pp. 41–125. Oxford University Press, 1994.

[2] L. Bachmair, T. Chen, C .R. Ramakrishnan, and I .V. Ramakrishnan. Subsumption algorithms based on search trees. In Proc. of CAAP'96, LNCS 1059, pp. 135–148, Springer Verlag, 1996.

[3] R. Caferra and N. Peltier. Decision procedures using model building techniques. In Hans Kleine Büning, editor, In Proc. of CSL'95, LNCS 1092, pp. 130–144, Springer Verlag, 1996.

[4] R. Caferra and N. Zabel. Extending resolution for model construction. In Proc. of JELIA'90, LNAI 478, pp. 153–169, Springer Verlag, 1991.

[5] A. Colmerauer. Equations and inequations on finite and infinite trees. In Proc. of FGCS'84, pp. 85–99, North-Holland, 1984.

[6] H. Comon and P. Lescanne. Equational problems and disunification. In *Journal of Symbolic Computation*, 7(3/4):371–425, 1989.

[7] U. Egly, R. Pichler and S. Woltran. On deciding subsumption problems, full paper. Available at: `http://www.kr.tuwien.ac.at/staff/stefan/qsat.pdf` (2002).

[8] C. G. Fermüller and A. Leitsch. Hyperresolution and automated model building. In *Journal of Logic and Computation*, 6(2):173–203, 1996.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[10] G. Gottlob, S. Marcus, A. Nerode, G. Salzer, and V. S. Subrahmanian. A non-ground realization of the stable and well-founded semantics. In *Theoretical Computer Science*, 166(1&2):221–262, 1996.

[11] G. Gottlob and R. Pichler. Working with ARMs: Complexity results on atomic representations of Herbrand models. *Information and Computation*, 165:183–207, 2001.

[12] G. Kuper, K. McAloon, K. Palem, and K. Perry. Efficient parallel algorithms for anti-unification and relative complement. In Proc. of LICS'88, pp. 112–120, IEEE Computer Society, 1988.

[13] D. Kapur, P. Narendran, D. Rosenkrantz, and H. Zhang. Sufficient-completeness, ground-reducibility and their complexity. In *Acta Informatica*, 28(4):311–350, 1991.

[14] K. Kunen. Answer sets and negation as failure. In Prof. of ICLP'87, pp. 219–228, MIT Press, 1987.

[15] J.-L. Lassez, M. Maher, and K. Marriott. Elimination of negation in term algebras. In Proc. of MFCS'91, LNCS 520, pp. 1–16, Springer Verlag, 1991.

[16] J.-L. Lassez and K. Marriott. Explicit representation of terms defined by counter examples. *Journal of Automated Reasoning*, 3(3):301–317, 1987.

[17] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, 1982.

[18] R. Pichler. On the complexity of H-subsumption. In Proc. of CSL'98, LNCS 1584, pp. 355–371, Springer Verlag, 1998.

[19] R. Pichler. Solving equational problems efficiently. In Proc. of CADE-16, LNAI 1632, pp. 97–111, Springer Verlag, 1996.

[20] R. Pichler. Completeness and redundancy in constrained clause logic. In *Automated Deduction in Classical and Non-Classical Logics, Selected Papers*, LNAI 1761, pp. 221–235, Springer Verlag, 2000.

[21] L. J. Stockmeyer. The Polynomial-Time Hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.