On unification for bounded distributive lattices

Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany
email: sofronie@mpi-sb.mpg.de

Abstract. We give a resolution-based procedure for deciding unifiability in the variety of bounded distributive lattices. The main idea is to use a structure-preserving translation to clause form to reduce the problem of testing the satisfiability of a unification problem $\mathcal{S}$ to the problem of checking the satisfiability of a set $\Phi_{\mathcal{S}}$ of (constrained) clauses. These ideas can be used for unification with free constants and for unification with linear constant restrictions. Complexity issues are also addressed.

1 Introduction

From an algebraic point of view, unification can be seen as solving (systems of) equations in the initial or free algebra of an equational theory. Apart from its theoretical interest, unification is used e.g. in resolution-based theorem proving and in term rewriting to deal with certain equational axioms (such as associativity and commutativity). The unification problem has been thoroughly studied for equationally defined theories characterized by axioms such as associativity, commutativity, distributivity, associativity-commutativity, associativity-commutativity-idempotency; and for several theories related to algebra (abelian groups, commutative and Boolean rings, semilattices, Boolean algebras, primal algebras, discriminator varieties). For details cf. [5] and the bibliography cited there. The combination of unification algorithms has been studied in [4].

In this paper we present some results on unification in the equational theory of bounded distributive lattices. The study was motivated, on the one hand, by our interest in distributive lattices with operators, and, on the other hand, by the fact that unification problems in semilattice- and lattice-based structures are becoming of increasing interest in computer science (we mention e.g. the results of Baader and Narendran on unification of concept terms in description logics [2]; similar possible applications in set constraints may also be of interest).

It is known that the class D_{01} of bounded distributive lattices has an undecidable first-order theory, but both its universal theory and its positive $\forall\exists$ theory (hence the unification problem with free constants) are decidable. Unification for distributive lattices has only been addressed in a few papers. In [12], Gerhard and Petrich give a criterion for unifiability (with free constants) of *two* terms in the theory of distributive lattices. (We were not able to generalize the argument used in the proof of this result to handle conjunctions of equations.) Then, in the attempt to give a basis set for all unifiers of two terms, they considered terms containing only one of the lattice operations $\vee$ or $\wedge$ and, for more general terms, only particular cases, containing few variables. The results of Ghilardi [13] show that the equational class D of distributive lattices has unification type 0, i.e. there exist D-unification problems with no minimal complete set of unifiers. We are not aware of any other results on unification for distributive lattices, e.g. concerning its complexity. Due to the interaction between operators, neither the ideas used in [19] for distributive unification, nor the results in [4] on the combination of unification algorithms can be applied in this case.

In [20], we gave a resolution-based decision procedure for the universal theory of certain varieties of distributive lattices with operators. The arguments in [20] cannot be used for the positive $\forall\exists$ theory of such varieties without modification. In this paper we further develop the ideas in [20] and show that the use of the Priestley representation for bounded distributive lattices allows us to give an algorithm based on resolution (with constrained clauses) for the unification problem in D_{01}. The algorithm consists of the following steps:

1. *Structure-preserving translation to clause form:* testing the satisfiability of a unification problem $\mathcal{S}$ is reduced to the problem of checking the satisfiability of a set $\Phi_{\mathcal{S}}$ of clauses. Expressing $\Phi_{\mathcal{S}}$ as a set of constrained clauses further simplifies the representation of the problem.
2. *Ordered resolution with selection* for (constrained) clauses is used for testing the satisfiability of $\Phi_{\mathcal{S}}$.

We also show that similar ideas can be used for unification with linear constant restrictions [4]. These results complete and improve the results in [12].

The main advantage of our approach is that the structure-preserving translation to clause form makes it much easier to treat the unification problem for bounded distributive lattices, by using results in resolution theory. As a byproduct, using Prop. 5.6 in [3], our results show that resolution (for ground clauses without equality) can be used for deciding the positive theory of D_{01}.

It seems that many of the results in this paper can be extended without difficulties to other varieties in which the free algebras have a description similar to those in D_{01}. This is the case for many subvarieties of the variety of Ockham algebras (bounded distributive lattices with a lattice antimorphism), such as, e.g., the variety of De Morgan algebras. For the sake of simplicity, in this paper we restrict our attention to the class of bounded distributive lattices only.

The paper is structured as follows. Section 2 contains the background information needed in the paper. Section 3 contains generalities about the unification problem for bounded distributive lattices. In Section 4 we give a resolution-based algorithm for this problem, and an extension to unification with linear constant restrictions. Section 5 contains conclusions and plans for future work.

2 Preliminaries

2.1 Algebra

Let Σ be a signature and $a : \Sigma \to \mathbb{N}$ an arity function. A Σ-algebra is a structure $\mathbf{A} = (A, \{\sigma_A\}_{\sigma \in \Sigma})$, where A is a non-empty set and for every $\sigma \in \Sigma$,

$\sigma_A : A^{a(\sigma)} \to A$. We denote by $T_\Sigma(X)$ the term algebra over Σ in the variables X. An *equation* is an expression of the form $t_1 = t_2$ where $t_1, t_2 \in T_\Sigma(X)$. A Σ-algebra $\mathbf{A} = (A, \{\sigma_A\}_{\sigma\in\Sigma})$ *satisfies* an equation $t_1 = t_2$ if t_1 and t_2 become equal for every substitution of elements in A for the variables. An equational class is the class of all algebras that satisfy a set of equations. If E is a set of equations in the signature Σ, then $F_\Sigma^E(X) := T_\Sigma(X)/{\equiv_E}$ is the free algebra over X in the equational class of all algebras that satisfy E (where $\equiv_E$ is the Σ-congruence on $T_\Sigma(X)$ generated by E). A *system of equations* is a finite set of equations $\mathcal{S} : \{s_1 = t_1, \ldots, s_k = t_k\}$, where $s_i, t_i \in T_\Sigma(X)$ for every $1 \leq i \leq k$. Let $\{y_1, \ldots, y_n\} \subseteq X$ be the set of all variables in $\mathcal{S}$. An algebra $\mathbf{A} = (A, \{\sigma_A\}_{\sigma\in\Sigma})$ satisfies the existential closure, $\exists y_1, \ldots y_n(s_1 = t_1 \wedge \cdots \wedge s_k = t_k)$, of $\mathcal{S}$ if there exists a map $h : X \to A$ such that $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$, where $\overline{h} : T_\Sigma(X) \to \mathbf{A}$ is the unique homomorphism of Σ-algebras that extends h.

2.2 Lattice theory

For the definition of partially-ordered set and order-filter we refer to [10]. If $\mathbf{X} = (X, \leq)$ is a partially-ordered set, we denote its set of order-filters by $\mathcal{O}(\mathbf{X})$. There is a bijective correspondence between $\mathcal{O}(\mathbf{X})$ and the set of all order-preserving maps from X to the partially-ordered set $\underline{\mathbf{2}} = (\{0,1\}, \leq)$, where $0 < 1$. A structure $\mathbf{L} = (L, \vee, \wedge)$, where L is a non-empty set and $\vee$ and $\wedge$ are two binary operations on L is a *lattice* if $\vee$ and $\wedge$ are associative, commutative and idempotent and satisfy the absorption laws. A *distributive lattice* is a lattice that satisfies either of the distributive laws. A lattice $\mathbf{L} = (L, \vee, \wedge)$ has a *first element* if there is an element $0 \in L$ such that $0 \leq x$ for every $x \in L$; it has a *last element* if there is an element $1 \in L$ such that $x \leq 1$ for every $x \in L$ (where $x \leq y$ iff $x \vee y = y$). A lattice having both a first and a last element is called *bounded*. In what follows, when we refer to bounded distributive lattices, the first and last element are supposed to be included in the signature. Thus, a bounded distributive lattice is a structure $\mathbf{L} = (L, \vee, \wedge, 0, 1)$, where $(L, \vee, \wedge)$ is a distributive lattice and $0, 1$ are constants such that 0 is first element and 1 last element in $(L, \vee, \wedge)$. We denote the equational class of all bounded distributive lattices by D_{01}. D_{01} contains e.g. the two-element bounded lattice, $\mathbf{2} = (\{0,1\}, \vee, \wedge, 0, 1)$, where $0 \vee 1 = 1, 0 \wedge 1 = 0$.

2.3 Priestley representation

If $\mathbf{L}$ is a bounded distributive lattice, let $D(\mathbf{L}) := \mathsf{Hom}_{\mathsf{D}_{01}}(\mathbf{L}, \mathbf{2})$ be set of all 0,1-lattice homomorphisms from $\mathbf{L}$ to the two-element bounded distributive lattice. The space $\mathbf{D}(\mathbf{L}) = (D(\mathbf{L}), \leq, \tau)$, where $\leq$ is the pointwise ordering on maps and τ is the topology generated by all sets of the form $X_a = \{h \in D(L) \mid h(a) = 1\}$ and their complements as a subbasis, is called the *Priestley dual* of $\mathbf{L}$. Let $\mathsf{Hom}_{\mathcal{P}}(\mathbf{D}(\mathbf{L}), \underline{\mathbf{2}})$ be the lattice of all continuous and order-preserving maps between the ordered topological space $\mathbf{D}(\mathbf{L})$, and the two-element partially-ordered set $\underline{\mathbf{2}}$ with the discrete topology. Priestley [18] showed that for every

$\mathbf{L} \in \mathsf{D}_{01}$, $\mathbf{L}$ is isomorphic to $\mathsf{Hom}_P(\mathbf{D}(\mathbf{L}), \underline{\mathbf{2}})$. In particular, if $\mathbf{L}$ is finite, then τ is the discrete topology, so $\mathbf{L}$ is isomorphic to $(\mathcal{O}(\mathbf{D}(\mathbf{L})), \cup, \cap, \emptyset, D(\mathbf{L}))$.

The dual of a finite distributive lattice is much smaller and less complex than the lattice itself. Therefore, problems concerning finite distributive lattices are likely to become simpler when translated into problems about their duals. We illustrate this by comparing the free algebra in D_{01} over a finite set C, $F_{\mathsf{D}_{01}}(C)$, and its Priestley dual $\mathbf{D}(F_{\mathsf{D}_{01}}(C))$. The theorem below is well-known.

Theorem 1. *Let C be a finite set. The following statements hold:*

(1) The map $p_C : (D(F_{\mathsf{D}_{01}}(C)), \leq) \to (2^C, \leq)$ defined for every $h \in D(F_{\mathsf{D}_{01}}(C))$ by $p_C(h) = h_{|C}$ (the restriction of $h : F_{\mathsf{D}_{01}}(C) \to \mathbf{2}$ to C) is an order-isomorphism, where in both cases the order is defined pointwise.
(2) The map $\eta_C : F_{\mathsf{D}_{01}}(C) \to \mathcal{O}(2^C, \leq)$ defined for every $t \in F_{\mathsf{D}_{01}}(C)$ by $\eta_C(t) = \{f : C \to \{0,1\} \mid \overline{f}(t) = 1\}$ (where for every $f : C \to \{0,1\}$, $\overline{f} : F_{\mathsf{D}_{01}}(C) \to \mathbf{2}$ is the unique extension of f to a 0,1-lattice homomorphism) is a lattice isomorphism. Its inverse is defined for every $U \in \mathcal{O}(2^C, \leq)$ by $\eta_C^{-1}(U) = \bigvee_{f \in U}(\bigwedge_{\{c \mid f(c)=1\}} c)$.

Every member of $F_{\mathsf{D}_{01}}(C)$ can be written as a finite join of finite meets of elements in C. Hence, $F_{\mathsf{D}_{01}}(C)$ is finite, and its number of elements is bounded by $2^{2^{|C|}}$. $|F_{\mathsf{D}_{01}}(C)|$ has been computed only for small values of $|C|$. By Theorem 1(1), $(D(F_{\mathsf{D}_{01}}(C)), \leq)$ is order-isomorphic to $(\mathcal{P}(C), \subseteq)$, hence has $2^{|C|}$ elements.

The main idea of this paper relies on this remark. The relatively simple structure of $D(F_{\mathsf{D}_{01}}(C))$ allows us to define a more efficient method for checking the satisfiability of unification problems with constants compared with methods that use the structure of $F_{\mathsf{D}_{01}}(C)$ and/or equational reasoning.

2.4 Unification

We present the definitions and results on E-unification needed in the paper.

Definition 1. *Let E be an equational theory, Σ its signature, and Δ a signature containing Σ. Let $\mathcal{S} : \{s_1 = t_1, \ldots, s_k = t_k\}$ be a system of equations, where $s_i, t_i \in T_\Delta(Y)$. Then $\mathcal{S}$ defines an E-*unification problem *over Δ. $\mathcal{S}$ is* elementary *iff $\Delta \subseteq \Sigma$; $\mathcal{S}$ is an E-unification problem with* (free) constants *iff $\Delta \backslash \Sigma$ is a set of constant symbols; and $\mathcal{S}$ is an E-unification problem with* linear constant restrictions *iff it is an E-unification problem with constants and, in addition, a linear ordering $<$ on the variables and free constants occurring in $\mathcal{S}$ is given. In a* general E-unification problem *$\Delta \backslash \Sigma$ may contain arbitrary function symbols.*

Definition 2. *A unification problem $\mathcal{S}$* has a solution *w.r.t. E if there is a substitution $\sigma : Y \to T_\Delta(Y)$ such that $\sigma(s_i) \equiv_E \sigma(t_i)$ for every $1 \leq i \leq k$.*

If $\mathcal{S}$ is an E-unification problem with linear constant restrictions, a solution for $\mathcal{S}$ is a substitution $\sigma : Y \to T_\Delta(Y)$ with the additional property that for every variable $y \in Y$ and every constant c, if $y < c$ then c does not occur in $\sigma(y)$.

In this context, one can study decidability of unifiability, the existence of unifiers, their classification according to "generality", or the possibility of determining minimal sets of unifiers which are complete, in the sense that all other unifiers are less general. In this paper we focus on testing unifiability. This is sufficient in many applications (e.g. in constraint-based approaches to automated deduction [9, 17, 16]) and is often simpler than computing complete sets of unifiers.

Theorem 2. *For any E-unification problem $\mathcal{S} : \{s_1 = t_1, \dots, s_k = t_k\}$ with free constants in C and variables $Y = \{y_1, \dots, y_m\}$ the following are equivalent:*

(1) $\mathcal{S}$ has a solution w.r.t. E.
(2) The formula $\exists y_1, \dots, y_m(s_1 = t_1 \wedge \dots \wedge s_k = t_k)$ is true in $F^E_{\Sigma \cup C}(\emptyset)$.
(3) There exists $h : Y \to F^E_\Sigma(C)$ such that $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$, where $\overline{h} : T_\Sigma(Y \cup C) \to F^E_\Sigma(C)$ is the unique extension of h to a homomorphism, such that, for all $c \in C$, $\overline{h}(c) = [c]$ (where $[c]$ is the equivalence class of c in $F^E_\Sigma(C)$).

Proof: (Sketch) The equivalence of (1) and (2) is proved e.g. in [7]. The equivalence of (2) and (3) follows from the fact that $U(F^E_{\Sigma \cup C}(Y))$ is isomorphic to $F^E_\Sigma(Y \cup C)$, where U maps a $\Sigma \cup C$-algebra to a Σ-algebra by forgetting the constants in the signature, i.e. $U((A, \{\sigma_A\}_{\sigma \in \Sigma \cup C})) = (A, \{\sigma_A\}_{\sigma \in \Sigma})$. $\square$

The importance of E-unification with linear constant restrictions is justified by the following theorem.

Theorem 3 ([5, 3]). *Let E be a non-trivial equational theory. The following statements are equivalent:*

(1) The positive theory[1] of E is decidable.
(2) General E-unification is decidable.
(3) E-unification with linear constant restrictions is decidable.

More precisely, as pointed out e.g. in [1], the decision problem for E-unification with linear constant restrictions can be reduced to the decision problem for general unification in linear time. The nondeterministic polynomial algorithm given in [3] can be used to reduce the decision problem for general unification to the decision problem for E-unification with linear constant restrictions.

3 Unification with constants in D_{01}. General remarks.

We now study the unification problem with free constants in the equational class D_{01} of bounded distributive lattices. We denote by D_{01} the equational theory of D_{01}. Let $\Sigma = \{\vee, \wedge, 0, 1\}$ be the signature of bounded distributive lattices. The following result is a direct consequence of Theorem 2.

[1] The positive theory of E is the collection of those closed formulae valid in the class of all models of E which are (equivalent to a formula) of the form $(Q_1 x_1) \dots (Q_m x_m)(\bigvee_{i=1}^q (s_{i1} = t_{i1} \wedge \dots \wedge s_{in_i} = t_{in_i}))$, where $Q_1, \dots, Q_m \in \{\exists, \forall\}$.

Corollary 1. *For any D_{01}-unification problem $\mathcal{S} : \{s_1 = t_1, \ldots, s_k = t_k\}$, with free constants in C and variables Y, the following are equivalent:*

(1) $\mathcal{S}$ has a solution w.r.t. D_{01}.
(2) There exists $h : Y \to F_{\mathsf{D}_{01}}(C)$ such that $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$, where $\overline{h} : T_\Sigma(Y \cup C) \to F_{\mathsf{D}_{01}}(C)$ is the unique extension of h to a homomorphism, such that $\overline{h}(c) = [c]$ for all $c \in C$.

By Corollary 1 and the fact that $F_{\mathsf{D}_{01}}(C)$ is finite for every finite C it follows that D_{01}-unification with free constants is decidable. This problem is co-NP hard: if $\mathcal{S}$ contains only one equation and no variables it reduces to the word problem for D_{01}, which has been shown to be co-NP hard [14].

We first present a straightforward (and rather inefficient) method for testing whether a unification problem has a solution. We then show how a simpler case (only one equation) is solved in [12]. In Section 4 we give a simpler method, which allows to test by resolution whether a unification problem has a solution.

The straightforward method. Let $\mathcal{S} : \{s_1 = t_1, \ldots, s_k = t_k\}$ be a D_{01}-unification problem with free constants in a finite set C and variables in the finite set Y. We can check if $\mathcal{S}$ has a solution by checking if there is an instantiation of the variables in $\mathcal{S}$ with elements in $F_{\mathsf{D}_{01}}(C)$ that satisfies $\mathcal{S}$. There exist at most $(2^{2^{|C|}})^{|Y|}$ such instantiations. For each instantiation $h : Y \to F_{\mathsf{D}_{01}}(C)$, one has to check if $h(s_i) \equiv_{D_{01}} h(t_i)$, $1 \leq i \leq k$. There exists an algorithm for disproving the equivalence of two terms which is nondeterministically polynomial in the length of the terms [14]. The elements in $F_{\mathsf{D}_{01}}(C)$ can be written as disjunctions of conjunctions of elements in C; the length of such a term is at most $|C| \cdot 2^{|C|}$. Hence, the length of $h(s_i)$ and $h(t_i)$, $1 \leq i \leq k$, can at most be $|Y| \cdot |C| \cdot 2^{|C|} + \mathsf{max}(\mathcal{S})$, where $\mathsf{max}(\mathcal{S})$ is the maximal length of a term occurring in $\mathcal{S}$.

A special case. In [12] Gerhard and Petrich present the following criterion for unifiability for one single equation, i.e. for the unification problem $\mathcal{S} : \{s = t\}$.

1. Let s' and t' be the disjunctive normal forms of s resp. t.
2. If neither s' or t' has a constant term[2] then s and t are unifiable.
3. If s' or t' has constant terms, let $h\colon Y \to T_{D_{01}}(C)$ be defined by $h(x) = D$ for every $x \in Y$, where D is the disjunction of all constant terms in s' and t'.
4. If $\overline{h}(s) \equiv_{D_{01}} \overline{h}(t)$ then s and t are unifiable, otherwise they are not unifiable.

The disjunction D in Step 3 can be determined in polynomial time w.r.t. $\mathsf{length}(s') + \mathsf{length}(t')$. The same holds for the process of replacing every variable in s and t by D. Both the length of D and the length of the result of replacing all variables in s, t by D ($\overline{h}(s)$ resp. $\overline{h}(t)$) is polynomial in $\mathsf{length}(s') + \mathsf{length}(t')$; but may be exponential in $\mathsf{length}(s) + \mathsf{length}(t)$. Hence, the complexity of the criterion above is given by the complexity of Step 1 (computing the disjunctive normal forms of s and t) and Step 4 (solving a word problem). The last problem is co-NP complete [14]; there exists an algorithm for disproving the equivalence of $\overline{h}(s)$ and $\overline{h}(t)$ which is nondeterministically polynomial in $\mathsf{length}(\overline{h}(s)) + \mathsf{length}(\overline{h}(t))$.

[2] If $s = \bigvee_i \bigwedge_{j \in I_i} s_j$ is in disjunctive normal form, then a constant term of s is any of the conjunctions $\bigwedge_{j \in I_i} s_j$ in s not containing any variable.

4 A resolution-based algorithm

We give a simpler algorithm for the problem of deciding whether a D_{01}-unification problem with free constants has a solution. The algorithm consists of two steps:

1. **Structure-preserving translation to clause form:**
   - Reduce the problem of testing the satisfiability of a unification problem $\mathcal{S}$ to checking the satisfiability of a set $\Phi_{\mathcal{S}}$ of clauses.
   - Show that $\Phi_{\mathcal{S}}$ can be expressed as a set of constrained clauses.
2. **Checking satisfiability by ordered resolution with selection:**
   - Use ordered resolution with selection for constrained sets of clauses to test the satisfiability of $\Phi_{\mathcal{S}}$.

4.1 Structure-preserving translation to clause form

In this section we reduce the problem of testing the satisfiability of a unification problem $\mathcal{S}$ to that of checking the satisfiability of a set of clauses. We do this in two steps: Theorem 4 shows that $F_{\mathsf{D}_{01}}(C)$ can be replaced with the lattice of order-filters of $(\mathcal{P}(C), \subseteq)$; Theorem 5 further reduces the problem to that of checking the satisfiability of a set of first-order (ground) clauses.

Theorem 4. *For any D_{01}-unification problem $\mathcal{S} : \{s_1 = t_1, \dots, s_k = t_k\}$ with free constants C and variables Y, the following are equivalent:*

(1) $\mathcal{S}$ has a solution w.r.t. D_{01}.
(2) There exists $h : Y \to F_{\mathsf{D}_{01}}(C)$ such that $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$, where $\overline{h} : T_\Sigma(Y \cup C) \to F_{\mathsf{D}_{01}}(C)$ is the unique homomorphism that extends h, such that $\overline{h}(c) = [c]$ for all $c \in C$.
(3) There exists $g : Y \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ such that $\overline{g}(s_i) = \overline{g}(t_i)$ for every $1 \leq i \leq k$, where $\overline{g} : T_\Sigma(Y \cup C) \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ is the unique homomorphism that extends g, such that $\overline{g}(c) = {\uparrow}\{c\} = \{X \subseteq C \mid c \in X\}$ for every $c \in C$.

Proof: (Idea) The equivalence of (1) and (2) follows directly from Corollary 1. The equivalence of (2) and (3) follows from the fact that there exists a 0,1-lattice isomorphism $\eta_C : F_{\mathsf{D}_{01}}(C) \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ defined for every $t \in F_{\mathsf{D}_{01}}(C)$ by $\eta_C(t) = \{f^{-1}(1) \cap C \mid f : F_{\mathsf{D}_{01}}(C) \to 2 \text{ is a 0,1-lattice homomorphism; } f(t) = 1\}$ such that for every $c \in C$, $\eta_C([c]) = {\uparrow}\{c\}$. □

Theorem 4 justifies a reduction of the problem of checking whether a unification problem with constants $\mathcal{S}$ has a solution to the problem of checking the satisfiability of a system of set constraints. This reduction can be then used to give a structure-preserving translation to clause form. Thus, the problem of checking whether a unification problem with constants $\mathcal{S}$ has a solution can be reduced to the problem of checking the satisfiability of a set of clauses.

The structure-preserving translation to clause form is inspired by Tseitin's well-known method for transforming quantifier-free formulae to clausal normal form and by the ideas in [20]. The link with set constraints mentioned above also explains the similarities with the structure-preserving translation to clause form presented in [6]. The remarks above are formally expressed by the following theorem.

Theorem 5. *Let $\mathcal{S} : \{s_1 = t_1, \dots, s_k = t_k\}$ be a D_{01}-unification problem with free constants C, and variables $Y = \{y_1, \dots, y_n\}$. Let $ST(\mathcal{S})$ be the set of all subterms of terms occurring in $\mathcal{S}$. The following are equivalent:*

(1) There exists $h : \{y_1, \dots, y_n\} \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ such that $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$, where $\overline{h} : T_\Sigma(Y \cup C) \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ is the unique homomorphism that extends h, such that $\overline{h}(c) = \uparrow\{c\}$ for every $c \in C$.

(2) There exists a family $\{I_e\}_{e \in ST(\mathcal{S})}$, such that $I_e \subseteq \mathcal{P}(C)$ for all $e \in ST(\mathcal{S})$, and for all $X, X_1, X_2 \subseteq C$ the following hold:

- *if $X_1 \in I_y$ and $X_1 \subseteq X_2$ then $X_2 \in I_y$, for every $y \in \{y_1, \dots, y_n\}$;*
- *$X \in I_{e_1 \wedge e_2}$ iff $X \in I_{e_1}$ and $X \in I_{e_2}$;*
- *$X \in I_{e_1 \vee e_2}$ iff $X \in I_{e_1}$ or $X \in I_{e_2}$;*
- *$I_0 = \emptyset$; $I_1 = C$; and for every $c \in C$, $X \in I_c$ iff $c \in X$;*
- *$X \in I_{s_i}$ iff $X \in I_{t_i}$ for all $1 \leq i \leq k$.*

(3) The conjunction of the following formulae is satisfiable:

(Her)		$P_y(X_1) \to P_y(X_2)$	*for all* $X_1 \subseteq X_2 \subseteq C, y \in \{y_1, \dots, y_n\}$
(Ren)	($\wedge$n)	$P_{e_1 \wedge e_2}(X) \to P_{e_i}(X)$	*for all* $X \subseteq C, i = 1, 2$
	($\wedge$p)	$P_{e_1}(X) \wedge P_{e_2}(X) \to P_{e_1 \wedge e_2}(X)$	*for all* $X \subseteq C$
	($\vee$n)	$P_{e_1 \vee e_2}(X) \to P_{e_1}(X) \vee P_{e_2}(X)$	*for all* $X \subseteq C$
	($\vee$p)	$P_{e_i}(X) \to P_{e_1 \vee e_2}(X)$	*for all* $X \subseteq C, i = 1, 2$
	(1)	$P_1(X)$	*for all* $X \subseteq C$
	(0)	$\neg P_0(X)$	*for all* $X \subseteq C$
	(cp)	$P_c(X)$	*for all* $X \subseteq C$ *with* $c \in X$
	(cn)	$\neg P_c(X)$	*for all* $X \subseteq C$ *with* $c \notin X$
(P)		$P_{s_i}(X) \leftrightarrow P_{t_i}(X)$,	*for all* $X \subseteq C$, *for all* $1 \leq i \leq k$

where each formula in (Her) $\cup$ (Ren) $\cup$ (P) *is the conjunction of all formulae obtained by instantiating the variables X, resp. X_1, X_2 with subsets of C satisfying the additional conditions; the indices $e_1 \vee e_2, e_1 \wedge e_2, 0, 1, c$ range over all elements in $ST(\mathcal{S})$; y ranges over all variables in $\{y_1, \dots, y_n\}$.*

Proof: (Sketch) (1) $\Rightarrow$ (2). For every $e \in ST(\mathcal{S})$ let $I_e := \overline{h}(e)$. Since $\overline{h}$ is a 0,1-homomorphism with $\overline{h}(c) = \uparrow\{c\}$, and the lattice operations in $\mathcal{O}(\mathcal{P}(C), \subseteq)$ are union and intersection, the family $\{I_e\}_{e \in ST(\mathcal{S})}$ satisfies the conditions in (2).

(2) $\Rightarrow$ (3) Let $\{I_e\}_{e \in ST(\mathcal{S})}$ be a family satisfying the conditions in (2). Then $(\mathcal{P}(C), \mathcal{I})$, where $\mathcal{I}(P_e) := I_e$ for all $e \in ST(\mathcal{S})$, is a model for (Her)$\cup$(Ren)$\cup$(P).

(3) $\Rightarrow$ (1) Assume that (Her)$\cup$(Ren)$\cup$(P) (a conjunction of ground clauses) is satisfied by the map $\mathcal{I} : \{P_e(X) \mid e \in ST(\mathcal{S}), X \subseteq C\} \to \{0, 1\}$. For every $y \in Y$ let $h(y) := \{X \in \mathcal{P}(C) \mid \mathcal{I}(P_y(X)) = 1\}$. Let $\overline{h} : T_\Sigma(Y \cup C) \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ be the unique homomorphism that extends h, such that $\overline{h}(c) = \uparrow\{c\}$ for every $c \in C$. As $\mathcal{I}$ satisfies (Her) $\cup$ (Ren), $\overline{h}(e) = \{X \in \mathcal{P}(C) \mid \mathcal{I}(P_e(X)) = 1\}$ for all $e \in ST(\mathcal{S})$. Since $\mathcal{I}$ satisfies (P), $\overline{h}(s_i) = \overline{h}(t_i)$ for every $1 \leq i \leq k$. $\square$

Corollary 2. *The D_{01}-unification problem $\mathcal{S} : \{s_1 = t_1, \dots, s_k = t_k\}$ with free constants C has a solution w.r.t. the equational theory of* D_{01} *iff the set of clauses* (Her) $\cup$ (Ren) $\cup$ (P) *is satisfiable.*

The satisfiability of (Her)$\cup$(Ren)$\cup$(P) can be checked for instance by resolution. We now give an upper bound on the complexity of deciding the satisfiability of (Her) $\cup$ (Ren) $\cup$ (P), i.e. for deciding the unifiability of $\mathcal{S}$.

Theorem 6. *(1) The problem of deciding whether the D_{01}-unification problem $\mathcal{S}$ has a solution can be solved in at most non-deterministically polynomial time in $|ST(\mathcal{S})|2^{|C|}$ (and in exponential time in $|ST(\mathcal{S})|2^{|C|}$ by using resolution).*

(2) If $\mathcal{S}$ only contains the operation symbols $\wedge, 0, 1$, and, possibly, constants, then the problem can be decided in at most polynomial time in $|ST(\mathcal{S})|2^{|C|}$.

Proof: Note first that the structure-preserving translation to clause form in Theorem 5 is polynomial in $|ST(\mathcal{S})|2^{|C|}$. The size of the conjunction of all formulae in (Her)$\cup$(Ren)$\cup$(P) is also polynomial in $|ST(\mathcal{S})|2^{|C|}$. (1) follows from this and the fact that the number of all distinct literals that can occur in the conjunction of ground clauses (Her)$\cup$(Ren)$\cup$(P) in Theorem 5(3) is bounded by $|ST(\mathcal{S})|2^{|C|}$. To prove (2) note that if only the operators $\wedge, 0, 1$ and, possibly, constants, occur in $\mathcal{S}$, then the clause form of (Her) $\cup$ (Ren) $\cup$ (P) is a set of ground Horn clauses. Dowling and Gallier [11] showed that satisfiability of a set Φ of ground Horn clauses can be proved in linear time w.r.t. the number of clauses in Φ. $\square$

Note. It is not necessary to explicitly add to (Her) $\cup$ (Ren) $\cup$ (P) formulae expressing the order relationship between the elements in $\mathcal{P}(C)$. However, these relationships have to be known when expressing (Her) $\cup$ (Ren) $\cup$ (P) as the conjunction of ground formulae by instantiating the variables with elements in $\mathcal{P}(C)$.

4.2 Translation to constrained clause form

The clause form of the set of formulae (Her)$\cup$(Ren)$\cup$(P) defined in Theorem 5(3) can be naturally expressed by constrained clauses of a special form as follows:

(Her)		$P_y(X_1) \rightarrow P_y(X_2)$	$[\![X_1 \subseteq X_2 \subseteq C]\!], y \in Y$
(Ren)	($\wedge$n)	$P_{e_1 \wedge e_2}(X) \rightarrow P_{e_i}(X)$	$[\![X \subseteq C]\!], i = 1, 2$
	($\wedge$p)	$P_{e_1}(X) \wedge P_{e_2}(X) \rightarrow P_{e_1 \wedge e_2}(X)$	$[\![X \subseteq C]\!]$
	($\vee$n)	$P_{e_1 \vee e_2}(X) \rightarrow P_{e_1}(X) \vee P_{e_2}(X)$	$[\![X \subseteq C]\!]$
	($\vee$p)	$P_{e_i}(X) \rightarrow P_{e_1 \vee e_2}(X)$	$[\![X \subseteq C]\!], i = 1, 2$
	(1)	$P_1(X)$	$[\![X \subseteq C]\!]$
	(0)	$\neg P_0(X)$	$[\![X \subseteq C]\!]$
	(cp)	$P_c(X)$	$[\![X \subseteq C, c \in X]\!]$
	(cn)	$\neg P_c(X)$	$[\![X \subseteq C, c \notin X]\!]$
(P)		$P_{s_i}(X) \leftrightarrow P_{t_i}(X)$,	$[\![X \subseteq C]\!]$, for all $1 \leq i \leq k$

Definition 3. *A* constrained clause *has the form $D[\![\phi]\!]$, where (i) D is a first-order clause with variables $X, X_1, \dots, X_n, \dots$ ranging over a countably infinite set V; all predicates occurring in D are unary; and (ii) the constraint ϕ is of the form $\bigwedge_{i \in I_1}(c_i \in X_i) \wedge \bigwedge_{i \in I_2}(c_i \notin X_i) \wedge \bigwedge_{i \in I_3, j \in I_4}(X_i \subseteq X_j) \wedge \bigwedge_{i \in I_1 \cup \dots \cup I_4}(X_i \subseteq C)$.*

Let $\mathcal{S}$ be a D_{01}-unification problem, and $\Phi_{\mathcal{S}}$ the set of constrained clauses associated with $\mathcal{S}$ as explained above. Then $\Phi_{\mathcal{S}}$ can be constructed in polynomial time in the size of $\mathcal{S}$. The size of $\Phi_{\mathcal{S}}$ is polynomial in the size of $\mathcal{S}$.

A substitution of the variables in V is called *ground* when it replaces every variable by an element of $\mathcal{P}(C)$ (this is the Herbrand universe of (Her) $\cup$ (Ren) $\cup$ (P)). A constrained clause $D[\![\phi]\!]$ represents the set $(D[\![\phi]\!])^g = \{D\sigma \mid \sigma \text{ ground}; \phi\sigma \text{ true}\}$ of all ground instances of D by instantiations of the variables which satisfy ϕ. We say that a set Φ of constrained clauses is satisfiable if the set of all its ground instances, $\Phi^g = \bigcup_{D[\![\phi]\!] \in N} (D[\![\phi]\!])^g$, is satisfiable.

4.3 A resolution calculus for constrained clauses

We now formulate a resolution calculus $\mathsf{CRes}_{\mathsf{S}}^{\succ}$ for the type of constrained clauses considered here. The calculus is parametrized by a total ordering $\succ$ on the predicate symbols, and a selection function S that assigns to each constrained clause $D[\![\phi]\!]$ a (possibly empty) multiset of (occurrences) of negative literals, called the selected literals of $D[\![\phi]\!]$. $\mathsf{CRes}_{\mathsf{S}}^{\succ}$ consists of the following rules[3].

Ordered resolution.

$$\frac{D_1 \vee P_e(X)\ [\![\phi_1(X, X_1, \ldots, X_n)]\!] \quad D_2 \vee \neg P_e(Z)\ [\![\phi_2(Z, Z_1, \ldots, Z_n)]\!]}{D_1 \vee D_2\sigma\ [\![\phi_1(X, X_1, \ldots, X_n) \wedge \phi_2(Z, Z_1, \ldots, Z_n)\sigma]\!]}$$

where $\sigma(Z) = X$ and $\sigma(W) = W$ in rest; P_e is the largest predicate symbol in $D_1 \vee P_e(X)$ and no literal is selected in $D_1 \vee P_e(X)$, and either $\neg P_e(Z)$ is selected in $D_2 \vee \neg P_e(Z)$, or otherwise nothing is selected in $D_2 \vee \neg P_e(Z)$ and P_e is the largest predicate symbol in $D_2 \vee \neg P_e(Z)$.

Positive factoring.

$$\frac{D \vee P_e(X) \vee P_e(Z)\ [\![\phi(X, Z, X_1, \ldots, X_n)]\!]}{D\sigma \vee P_e(X)\ [\![\phi(X, Z, X_1, \ldots, X_n)\sigma]\!]}$$

where $\sigma(Z) = X$, and $\sigma(W) = W$ in rest; P_e is the largest predicate symbol in $D \vee P_e(X) \vee P_e(Z)$ and nothing is selected in $D \vee P_e(X) \vee P_e(Z)$.

Theorem 7. *Let Φ be a set of constrained clauses, $\succ$ a total order on the predicate symbols, and S a selection function. Φ is unsatisfiable iff the empty clause $\Box[\![\phi]\!]$ (constrained by a satisfiable constraint ϕ) is derivable from Φ in $\mathsf{CRes}_{\mathsf{S}}^{\succ}$.*

Proof: (Idea) The proof uses the completeness of ordered resolution with selection for ground clauses and a lifting lemma for constrained clauses; the arguments are similar to those in [9]. $\Box$

[3] The ordered resolution calculus with selection $\mathsf{Res}_{\mathsf{S}}^{\succ}$ is complete for any well-founded and total ordering $\succ$ on ground literals and any selection function S. Here we consider a less restrictive form of resolution for constrained clauses, in order to simplify the presentation by avoiding the necessity of also handling order constraints (w.r.t. $\succ$).

Example: Decide whether the D_{01}-unification problem $\mathcal{S} : \{y \wedge c = 0, y \vee c = 1\}$ has a solution, where c is a constant and y a variable.
(Note that $\mathcal{S}$ corresponds to the formula: $\forall c \exists y (y \wedge c = 0$ and $y \vee c = 1.)$

Solution: Let $\succ$ be a total ordering on the predicate symbols, defined such that $P_e \succ P_{e'}$ whenever (i) e' is a subterm of e; or (ii) e is a non-atomic term and c a constant; or (iii) e is a non-atomic term or a constant and x is a variable. Let S be a selection function that selects all negative occurrences of literals except in $\mathsf{Ren}(\vee\mathsf{p}, \wedge\mathsf{p})$, where nothing is selected. By the structure-preserving translation to clause form in Theorem 5(3) we obtain the following set of constrained clauses:

$$
\begin{array}{lrl}
(1) & \underline{P_y(X)} \rightarrow P_y(Y) & [\![X \subseteq Y \subseteq C]\!] \\
(2) & \underline{P_{y\wedge c}(X)} \rightarrow P_y(X) & [\![X \subseteq C]\!] \\
(3) & \underline{P_{y\wedge c}(X)} \rightarrow P_c(X) & [\![X \subseteq C]\!] \\
(4) & P_y(X) \wedge P_c(X) \rightarrow \boxed{P_{y\wedge c}(X)} & [\![X \subseteq C]\!] \\
(5) & \underline{P_{y\vee c}(X)} \rightarrow P_y(X) \vee P_c(X) & [\![X \subseteq C]\!] \\
(6) & P_y(X) \rightarrow \boxed{P_{y\vee c}(X)} & [\![X \subseteq C]\!] \\
(7) & P_c(X) \rightarrow \boxed{P_{y\vee c}(X)} & [\![X \subseteq C]\!] \\
(8) & \neg P_0(X) & [\![X \subseteq C]\!] \\
(9) & P_1(X) & [\![X \subseteq C]\!] \\
(10) & P_c(X) & [\![X \subseteq C, c \in X]\!] \\
(11) & \neg P_c(X) & [\![X \subseteq C, c \notin X]\!] \\
(12) & \underline{P_{y\wedge c}(X)} \rightarrow P_0(X) & [\![X \subseteq C]\!] \\
(13) & \underline{P_0(X)} \rightarrow P_{y\wedge c}(X) & [\![X \subseteq C]\!] \\
(14) & \underline{P_{y\vee c}(X)} \rightarrow P_1(X) & [\![X \subseteq C]\!] \\
(15) & \underline{P_1(X)} \rightarrow P_{y\vee c}(X) & [\![X \subseteq C]\!]
\end{array}
$$

where $C = \{c\}$, the selected literals are underlined and the positive literals containing the maximal predicate symbol are in boxes. All ground inferences of (13) and (14) are redundant (so, (13) and (14) can be considered to be redundant). We obtain the following deduction of the empty clause $\square$:

$$
\begin{array}{lll}
(16)\ \underline{P_y(X)} \wedge \underline{P_c(X)} \rightarrow P_0(X) & [\![X \subseteq C]\!] & \text{(by (12) and (4))} \\
(17)\ \underline{P_y(X)} \rightarrow P_0(X) & [\![c \in X, X \subseteq C]\!] & \text{(by (10) and (16))} \\
(18)\ P_{y\vee c}(X) & [\![X \subseteq C]\!] & \text{(by (15) and (9))} \\
(19)\ P_y(X) \vee \boxed{P_c(X)} & [\![X \subseteq C]\!] & \text{(by (18) and (5))} \\
(20)\ P_y(X) & [\![c \notin X, X \subseteq C]\!] & \text{(by (19) and (11))} \\
(21)\ P_y(Y) & [\![c \notin X, X \subseteq Y \subseteq C]\!] & \text{(by (20) and (1))} \\
(22)\ P_0(X) & [\![c \notin X, c \in X, X \subseteq C]\!] & \text{(by (20) and (17))} \\
(23)\ \square & [\![c \notin X, c \in X, X \subseteq C]\!] & \text{(by (22) and (8))} \\
(24)\ P_0(Y) & [\![c \notin X, c \in Y, X \subseteq Y \subseteq C]\!] & \text{(by (21) and (17))} \\
(25)\ \square & [\![c \notin X, c \in Y, X \subseteq Y \subseteq C]\!] & \text{(by (21) and (17))}
\end{array}
$$

The constraint in (23) is unsatisfiable, but the constraint in (25) is satisfiable (e.g. by $X = \emptyset$ and $Y = C$)[4]. So, the set consisting of the clauses (1)–(15) is unsatisfiable, hence $\mathcal{S}$ has no solution.

[4] This shows that inferences with the clause Her in Theorem 5(3) (in particular, with clause (1) for this example) are necessary for the correctness of the method.

4.4 Complexity considerations for some special cases.

We now analyze some situations in which deciding D_{01}-unifiability is especially easy. We start by showing that for unification problems of the form $\mathcal{S} : \{s = t\}$ the algorithm performs well. We end by analyzing the complexity of unification without constants.

Unification with free constants: general case. Let $\mathcal{S}$ be a unification problem. Let $\succ$ be a total ordering on the predicate symbols $\{P_e \mid e \in ST(\mathcal{S})\}$, defined such that $P_e \succ P_{e'}$ whenever (i) e' is a subformula of e; or (ii) e is a non-atomic formula and c a constant; or (iii) e is a non-atomic formula or a constant and y is a variable. Let S be a selection function that (i) selects nothing in $\mathsf{Ren}(\vee\mathsf{p}, \wedge\mathsf{p})$, and (ii) in every other non-positive clause selects the set of all occurrences of negative literals that contain the maximal predicate symbol(s) among those occurring in the negative literals in the clause. Then:

1. no inferences are possible between (Her) and (Ren);
2. all inferences between two clauses in (Ren) generate tautologies;
3. no inferences are possible between (Her) and clauses in (P);
4. inferences between $P_{s_i}(X) \rightarrow P_{t_i}(X)$ in (P) and clauses in (Ren) lead to:
   (a) $\bigwedge_{j \in J} P_{e^i_j}(X) \rightarrow P_{t_i}(X)$ $[\![X \subseteq C]\!]$, where for every J, $\{e^i_j \mid j \in J\}$ is a multiset of subterms of $ST(s_i)$, containing no repetition of subterms that occur at the same position in s_i;
   (b) $(\bigwedge P_{c_j}(X)) \wedge (\bigwedge P_{x_l}(X)) \rightarrow P_{t_i}(X)$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$;
   (c) $\bigwedge P_{x_l}(X) \rightarrow P_{t_i}(X)$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$;
   (d) $P_{t_i}(X)$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$.
   There are at most $2^{\mathsf{length}(s_i)}$ such clauses; all constraints are linear in $\mathsf{length}(s_i)$.
5. inferences between $P_{t_i}(X)$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$ and clauses in (Ren) lead to:
   (a') $\bigvee_{j \in J} P_{e^i_j}(X)$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$, where for every J, $\{e^i_j \mid j \in J\}$ is a multiset of subterms of $ST(t_i)$, containing no repetition of subterms that occur at the same position in t_i;
   (b') $(\bigvee P_{c_j}(X)) \vee (\bigwedge P_{x_l}(X))$ $[\![X \subseteq C, d_i \in X, i \in I, d'_k \notin X, k \in K]\!]$;
   (c') $\bigvee P_{x_l}(X)$ $[\![X \subseteq C, d_i \in X, i \in I, d'_k \notin X, k \in K]\!]$;
   (d') $\Box$ $[\![X \subseteq C, d_i \in X, i \in I]\!]$.
   or factors thereof. There are at most $2^{\mathsf{length}(t_i)}$ such clauses; all constraints are linear in $\mathsf{length}(s_i) + \mathsf{length}(t_i)$.
6. inferences between clauses of type (c') and (Her) produce clauses of the form.
   (e') $\bigvee P_{x_l}(X_i)$ $[\![X \subseteq C, d_i \in X, i \in I, d'_k \notin X, k \in K, X \subseteq X_i]\!]$;
7. further inferences may be possible between clauses of the form (a)–(c) and clauses of type (d) or (a')–(c') and (e') and resolvents thereof. (These inferences can be further controlled, e.g. by adopting an additional labeling of the predicate symbols P_e that also indicates in which of the terms $t_1, \dots, t_k, s_1, \dots, s_k$ and at which position e occurs; and by defining redundancy criteria. Here, we do not enter in further details.)

Corollary 3. *If all clauses generated from* $(\mathsf{Her}) \cup (\mathsf{Ren}) \cup (\mathsf{P})$ *by* $\mathsf{CRes}^{\succ}_{\mathsf{S}}$ *contain a negative literal then* $\mathcal{S}$ *has a solution.*

Proof: Follows from the remarks above and the fact that if all clauses generated contain a (selected) negative literal then $\Box$ cannot be obtained by $\mathsf{CRes}^{\succ}_{\mathsf{S}}$. $\Box$

This happens e.g. if for all $1 \leq i \leq k$, the disjunctive normal forms of s_i, t_i do not contain constant terms.

Unification with free constants: one equation. If $\mathcal{S}$ consists of only one equation, the last type of inferences can be proved to produce only constrained clauses with the property that all their ground instances are already subsumed by the set of ground instances of clauses of type (d) previously generated, as will be shown in Corollary 4.

Lemma 1. *The satisfiability of a constraint ϕ of size $|\phi|$ can be checked in time linear in $|C| \cdot |\phi|$.*

Proof: (Sketch) Note first that a constraint

$$\phi = \bigwedge_{i \in I_1} (c_i \in X_i) \wedge \bigwedge_{i \in I_2} (d_i \notin X_i) \wedge \bigwedge_{i \in I_3, j \in I_4} (X_i \subseteq X_j) \wedge \bigwedge_{i \in I_1 \cup \dots \cup I_4} (X_i \subseteq C)$$

is satisfiable iff the set

$$E_\phi = \{P_{X_i}(c_i) \mid i \in I_1\} \cup \{\neg P_{X_j}(d_j) \mid j \in I_2\} \cup \{P_{X_i}(x) \rightarrow P_{X_j}(x) \mid i \in I_3, j \in I_4\}$$

of Horn clauses is satisfiable by a Herbrand interpretation. Moreover, E_ϕ is satisfiable iff the set G_ϕ of all its ground instances (which has cardinality $|C| \times |\phi|$) is satisfiable. Since all clauses in G_ϕ are ground Horn clauses, it follows by results in [11] that the satisfiability of G_ϕ can be checked in time linear in the size of G_ϕ i.e. linear in $|C| \times |\phi|$. $\Box$

Corollary 4. *Let $\mathcal{S} : \{s = t\}$ consist only of one equation. Then the satisfiability of $\mathcal{S}$ can be checked in time $2^{\mathsf{length}(s)+\mathsf{length}(t)}(\mathsf{length}(s) + \mathsf{length}(t))$.*

Proof: (Idea) In a first step, by inferences with (Ren), $P_s(X) \rightarrow P_t(X)$ generates literals of the form (a)–(c), and, possibly of type (d). Let $P_t(X)[\![\phi_i]\!], i \in I$ be all clauses of type (d) generated this way. Similarly, let $P_s(X)[\![\phi'_j]\!], j \in J$ be all clauses of type (d) generated from $P_t(X) \rightarrow P_s(X)$. By the remarks at the beginning of this subsection, ϕ_i, ϕ_j contain only constraints of the form $X \subseteq C$ and $c_i \in X$. By inferences between $P_t(X)[\![\phi_i]\!], i \in I$ and $P_t(Z) \rightarrow P_s(Z)$ $[\![Z \subseteq C]\!]$ the clause $P_s[\![\phi_i]\!]$ is generated. $P_t[\![\phi'_j]\!]$ are generated, for all $j \in J$, in a similar way. The constraint of a clause of type (a')–(d') contains, as a conjunct, one of the constraints ϕ_i or ϕ'_j. Let $D_i \rightarrow P_t(X)[\![\psi]\!]$ be one of the clauses in (a)–(c). A resolution with a clause of type (a')–(d') would produce a clause of the form $D'_i \rightarrow P_t[\![\psi \wedge \phi_i \wedge \rho]\!]$. The set of ground instances of such a clause is redundant with respect any set of clauses that contains all ground instances of P_t $[\![\phi_i]\!]$, hence (by the proof of Theorem 7) all such clauses can be considered redundant.

Similar arguments concerning redundancy of generated clauses can be used to control the inferences with (Her) and to prove that all resolvents of clauses of type (a')–(d') as well as resolvents of inferences with Her and clauses of type (a)–(d) have the property that all their ground instances are subsumed by ground instances of clauses of type (d). The conclusion follows since $2^{|\mathcal{S}|}$ clauses are generated this way and the constraints can be checked in linear time. $\square$

Unification without constants in D_{01}. If $C = \emptyset$, then $F_{\mathsf{D}_{01}}(C)$ is the two-element lattice. By Theorem 2, a D_{01}-unification problem $\mathcal{S} : \{s_1 = t_1, \dots, s_k = t_k\}$ with variables $\{y_1, \dots, y_n\}$ and no constants has a solution w.r.t. the equational theory of D_{01} iff the existential closure of $\mathcal{S}$, $\exists y_1, \dots y_n(s_1 = t_1 \wedge \dots \wedge s_k = t_k)$ is valid in the two-element lattice. The number of clauses corresponding to $(\mathsf{Her}) \cup (\mathsf{Ren}) \cup (\mathsf{P})$ in Theorem 5 is in this case polynomial in $ST(\mathcal{S})$.

Theorem 8 (Complexity). *Let $\mathcal{S}$ be a D_{01}-unification problem without constants. Assume that all terms in $\mathcal{S}$ have been simplified by (recursively) applying the following simplification rules*[5]*: $e \wedge 1 \mapsto e; e \wedge 0 \mapsto 0; e \vee 1 \mapsto 1; e \vee 0 \mapsto e$.*

(1) If $\{0, 1\} \not\subseteq ST(\mathcal{S})$, then $\mathcal{S}$ always has a solution.
(2) If $\{0, 1\} \subseteq ST(\mathcal{S})$ and $\mathcal{S}$ consists of only one equation (or else it contains the equation $1 = 0$) then $\mathcal{S}$ has no solution.
(3) If $\mathcal{S}$ only contains the operators $\wedge, 0, 1$, then the problem of checking whether $\mathcal{S}$ has a solution can be solved in polynomial time. The same holds if $\mathcal{S}$ only contains the operators $\vee, 0, 1$.
(4) Otherwise, the problem of checking whether $\mathcal{S}$ has a solution is NP-complete.

Proof: (Sketch) (1) Assume that $1 \notin ST(\mathcal{S})$. Let $\Phi_{\mathcal{S}}$ be the set of clauses associated to S by the structure-preserving translation to clause form in Theorem 5(3). Since no constant occurs in $\mathcal{S}$, all the clauses $\Phi_{\mathcal{S}}$ are non-positive, so $\Phi_{\mathcal{S}}$ is satisfiable (consider a selection function that selects all negative literals in all clauses; then no resolution inference is possible). The case $0 \notin ST(\mathcal{S})$ follows by duality.

(2) is obvious, and (3) follows from the second part of Theorem 6.

(4) The problem of deciding whether a unification problem without constants has a solution is clearly in NP. NP-hardness follows from the fact that the satisfiability problem for Boolean formulae of the form $E = F \wedge \neg G$, where F and G only contain the operators $\vee$ and $\wedge$ (which is NP-complete [15]) can be reduced in polynomial time to the satisfiability of a D_{01}-unification problem with at least two equations $s_i = 0$ and $s_j = 1$, namely $\mathcal{S} : \{F = 1, G = 0\}$. $\square$

4.5 Unification with linear constant restrictions

We now show that the idea used in the method described in Section 4.1 can be adapted to give an algorithm for unification with linear constant restrictions.

We first express the fact that $t \in F_{\mathsf{D}_{01}}(C \backslash \{c\})$ by using the isomorphism $\eta_C : F_{\mathsf{D}_{01}}(C) \to \mathcal{O}(\mathcal{P}(C), \subseteq)$ defined for every $t \in F_{\mathsf{D}_{01}}(C)$ by $\eta_C(t) = \{f^{-1}(1) \cap C \mid f : F_{\mathsf{D}_{01}}(C) \to 2$ is a 0,1-lattice homomorphism; $f(t) = 1\}$ (cf. Theorem 1).

Lemma 2. *If $t \in F_{\mathsf{D}_{01}}(C)$ then there exists $t' \in F_{\mathsf{D}_{01}}(C \backslash \{c\})$ such that $t \equiv_{\mathsf{D}_{01}} t'$ iff $\eta_C(t) = \emptyset$ or $C \backslash \{c\} \in \eta_C(t)$.*

[5] This can be done in polynomial time.

Proof: (Idea) This is a consequence of the fact that for every $t \in F_{\mathsf{D}_{01}}(C)$, there exists $t' \in F_{\mathsf{D}_{01}}(C\backslash\{c\})$ such that $t \equiv_{\mathsf{D}_{01}} t'$ iff for every $X \subseteq C$, if $c \in X$ then either $X \notin \eta_C(t)$ or X is not minimal in $\eta_C(t)$. The proof of the equivalence above uses the way η_D is defined for every D, and the fact that there exists an injective homomorphism $i : F_{\mathsf{D}_{01}}(C\backslash\{c\}) \to F_{\mathsf{D}_{01}}(C)$, such that for every $U \in \mathcal{O}(\mathcal{P}(C\backslash\{c\}))$, $\eta_C(i(\eta^{-1}_{C\backslash\{c\}}(U)))$ is the order-filter generated by U in $(\mathcal{P}(C), \subseteq)$. □

As in the case of unification with free constants, the remark above justifies a structure-preserving translation to clause form.

Theorem 9. *Let $\mathcal{S} : \{s_1 = t_1, \ldots, s_k = t_k\}$ be a D_{01}-unification problem with linear constant restrictions Lcr, constants C and variables Y. The following are equivalent:*

(1) $\mathcal{S}$ has a solution w.r.t. the equational theory of D_{01}.
(2) The conjunction of the following set of formulae is satisfiable:

$$\begin{array}{lll}
\textsf{(Her)} & P_y(X_1) \to P_y(X_2) & \textit{for all } X_1 \subseteq X_2 \subseteq C, y \in Y \\
\textsf{(Ren)} & P_{e_1 \wedge e_2}(X) \leftrightarrow P_{e_1}(X) \wedge P_{e_2}(X) & \textit{for all } X \subseteq C \\
 & P_{e_1 \vee e_2}(X) \leftrightarrow P_{e_1}(X) \vee P_{e_2}(X) & \textit{for all } X \subseteq C \\
 & P_1(X) & \textit{for all } X \subseteq C \\
 & \neg P_0(X) & \textit{for all } X \subseteq C \\
 & P_c(X) & \textit{for all } X \subseteq C \textit{ with } c \in X \\
 & \neg P_c(X) & \textit{for all } X \subseteq C \textit{ with } c \notin X \\
\textsf{(Lcr)} & P_y(X) \to P_y(C\backslash\{c\}) & \textit{for all } X \subseteq C \textit{ if } y < c \in Lcr \\
\textsf{(P)} & P_{s_i}(X) \leftrightarrow P_{t_i}(X), & \textit{for all } 1 \leq i \leq k
\end{array}$$

Proof: (Sketch) This follows from Definition 2, Lemma 2, and arguments similar to those used for Theorem 5. □

5 Conclusion

We presented a resolution-based method for deciding unifiability w.r.t. the equational theory of bounded distributive lattices with operators. The method uses the Priestley representation for bounded distributive lattices, in particular the description of the dual $\mathbf{D}(F_{D_{01}}(C))$ of the free lattice in D_{01} over C as $(\mathcal{P}(C), \subseteq)$.

This helped us to reduce the problem of checking whether a D_{01}-unification problem $\mathcal{S}$ with constants C (and linear constraint restrictions) has a solution, to the problem of checking the satisfiability of a set $\Phi_{\mathcal{S}}$ of clauses. $\Phi_{\mathcal{S}}$ can be represented both as a finite set of ground clauses, and as a set of constrained clauses; the last representation is much more compact. We formulated a resolution calculus for such constrained clauses and proved its soundness and completeness. The method we give is in general still exponential in $|ST(\mathcal{S})|2^C$. However, in several situations it is more efficient than other existing methods: syntactic information about the terms in $\mathcal{S}$ is sometimes reflected by the form of clauses, which allows us to establish better upper bounds for these particular problems. Our algorithm also behaves well for unification problems consisting of only one equation.

It would be interesting to compare our method with more general unification algorithms, e.g. based on rewriting. One such algorithm [8] decides unifiability *over* the free algebra, i.e. in an algebraic extension of the free algebra. We analyzed this more general problem for the equational theory of D_{01}. As part of work in progress, we prove that this reduces to Boolean unifiability (due to the fact that the algebraically closed elements of D_{01} are the Boolean lattices).

Acknowledgments. I thank H. Ganzinger for drawing my attention to the results on unification with linear constant restrictions (cf. e.g. [5]) and on Boolean unification [1], and to the possibility of using constrained clauses (for theorem proving for many-valued logics), an idea which proved to be useful in this paper. I thank J.M. Talbot for the discussion we had on possible applications of unification to set constraints.

References

1. F. Baader. On the complexity of Boolean unification. *Information Processing Letters*, 67(4):215–220, 1998.
2. F. Baader and P. Narendran. Unification of concept terms in description logics. In H. Prade, editor, *Proceedings of ECAI'98*, pages 331–335. Wiley, 1998.
3. F. Baader and K.U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symbolic Computation*, 21:211–243, 1996.
4. F. Baader and K.U. Schulz. Combination of constraint solvers for free and quasi-free structures. *Theoretical Computer Science*, 192:107–161, 1998.
5. F. Baader and W. Snyder. Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning.* Elsevier, 2000. To appear.
6. L. Bachmair, H. Ganzinger, and U. Waldmann. Set constraints are the monadic class. In *Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 75–83, Montreal, Canada, June 19–23, 1993. IEEE Computer Society Press, Los Alamitos, CA, USA.
7. A. Bockmayr. Algebraic and logical aspects of unification. In K.U. Schulz, editor, *Proceedings of Word Equations and Related Topics IWWERT'90, Tübingen, October 1990*, LNCS 572, pages 171–180. Springer Verlag, 1992.
8. A. Bockmayr. Model-theoretic aspects of unification. In K.U. Schulz, editor, *Proceedings of Word Equations and Related Topics IWWERT'90, Tübingen, October 1990*, LNCS 572, pages 181–196. Springer Verlag, 1992.
9. H.J. Bürckert. *A resolution principle for a logic with restricted quantifiers.* LNAI 568. Springer Verlag, 1991.
10. S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra.* Graduate Texts in Mathematics. Springer, 1981.
11. W.F. Dowling and J.H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming*, 1(3):267–284, 1984.
12. J.A. Gerhard and M. Petrich. Unification in free distributive lattices. *Theoretical Computer Science*, 126(2):237–257, 1994.
13. S. Ghilardi. Unification through projectivity. *J. Logic and Computation*, 7(6):733–752, 1997.
14. H.B. Hunt, D.J. Rosenkrantz, and P.A. Bloniarz. On the computational complexity of algebra of lattices. *SIAM Journal of Computation*, 16(1):129–148, 1987.

15. H.B. Hunt and R.E. Stearns. The complexity of very simple Boolean formulas with applications. *SIAM J. Comput.*, 19(1):44–70, 1990.
16. C. Kirchner and H. Kirchner. Constrained equational reasoning. In *UNIF'89 Extended Abstacts of the 3rd Int. Workshop on Unification*, pages 160–171, Pfalzakademie, Lambrecht, 1989.
17. R. Nieuwenhuis and A. Rubio. Theorem proving with ordering constrained clauses. In D. Kapur, editor, *Proceedings of CADE-11*, LNAI 607, pages 477–491. Springer Verlag, 1992.
18. H.A. Priestley. Ordered topological spaces and the representation of distributive lattices. *Proc. London Math. Soc.*, 3:507–530, 1972.
19. M. Schmidt-Schauß. A decision algorithm for distributive unification. *Theoretical Computer Science*, 208(1–2):111–148, 1998.
20. V. Sofronie-Stokkermans. On the universal theory of varieties of distributive lattices with operators: Some decidability and complexity results. In H. Ganzinger, editor, *Proceedings of CADE-16*, LNAI 1632, pages 157–171. Springer Verlag, 1999.