Decidability Results for Saturation-Based Model Building

Matthias Horbach
and Christoph Weidenbach

MPI–I–2009–RG1–004 Dez. 2009

Authors' Addresses

Matthias Horbach and Christoph Weidenbach
Max Planck Institute for Informatics
Campus E1 4
66123 Saarbrücken
Germany

Publication Notes

This report is a preliminary version of an article intended for publication elsewhere.

Acknowledgements

We thank Peter Baumgartner for valuable background information.

Matthias Horbach and Christoph Weidenbach are supported by the German Transregional Collaborative Research Center SFB/TR 14 AVACS.

Abstract

Saturation-based calculi such as superposition can be successfully instantiated to decision procedures for many decidable fragments of first-order logic. In case of termination without generating an empty clause, a saturated clause set implicitly represents a minimal model for all clauses, based on the underlying term ordering of the superposition calculus. In general, it is not decidable whether a ground atom, a clause or even a formula holds in this minimal model of a satisfiable saturated clause set.

Based on an extension of our superposition calculus for fixed domains with syntactic disequality constraints in a non-equational setting, we describe models given by ARM (Atomic Representations of term Models) or DIG (Disjunctions of Implicit Generalizations) representations as minimal models of finite saturated clause sets. This allows us to present several new decidability results for validity in such models. These results extend in particular the known decidability results for ARM and DIG representations.

Keywords

decidability, ordered resolution, model building

Contents

1 Introduction

Saturation-based calculi such as ordered resolution [2] or superposition [18] can be successfully instantiated to decision procedures for many decidable fragments of first-order logic [10, 16, 13]. Given a set N of clauses, saturation means the exhaustive recursive application of all calculus inference rules up to redundancy, resulting in a potentially infinite clause set N^*. If the calculus is complete, either N^* contains the empty clause, meaning that N is unsatisfiable, or the set N^* implicitly represents a unique minimal Herbrand model $\mathcal{I}_{N^*}$ produced by a model generating operator out of N^*. The model generating operator is based on a reduction ordering $\prec$ that is total on ground terms and also used to define the redundancy notion and inference restrictions underlying a superposition calculus.

Given a model representation formalism for some clause set N, according to [9, 4], each model representation should ideally represent a *unique* single interpretation, provide an *atom test* deciding ground atoms, support a *formula evaluation* procedure deciding arbitrary formulae, and an algorithm deciding *equivalence* of two model representations.

By definition, the superposition model generating operator produces a unique minimal model $\mathcal{I}_{N^*}$ out of N^* according to $\prec$. This satisfies the above uniqueness postulate. As first-order logic is semi-decidable, the saturated set N^* may be infinite and hence decision procedures for properties of $\mathcal{I}_{N^*}$ are in general hard to find. Even if N^* is finite, any other properties like the ground atom test, formula evaluation, and equivalence of models are still undecidable in general. This shows the expressiveness of the saturation concept. For particular cases, however, more is known. The atom test is decidable if N^* is finite and all clauses $\Gamma \rightarrow \Delta, s{\approx}t \in N^*$ are universally reductive, i.e., $\text{vars}(\Gamma, \Delta, t) \subseteq \text{vars}(s)$ and s is the strictly maximal term in $\Gamma \rightarrow \Delta, s{\approx}t$ or a literal in Γ is selected [12]. This basically generalizes the well-known decidability result of the word problem for convergent rewrite systems to full clause representations. Even for a finite universally reductive clause set N^*, clause evaluation (and therefore formula evaluation) and the model equivalence of two such clause sets remain undecidable.

More specific resolution strategies produce forms of universally reductive saturated clause sets with better decidability properties. An eager selection strategy results in a hyper-resolution style saturation process where, starting with a Horn clause set N, eventually all clauses contributing to the model $\mathcal{I}_{N^*}$ are positive units. Such strategies decide, e.g., the clause classes $\mathcal{VED}$ and $\mathcal{PVD}$ [9, 4]. The positive unit clauses in N^* represent so-called ARMs (Atomic Representations of term Models). Saturations of resolution calculi with constraints [18, 4] produce in a similar setting positive unit clauses with constraints. Restricted to syntactic disequality constraints, the minimal model of the saturated clause set can be represented as a DIG (Disjunctions of Implicit Generalizations). DIGs generalize ARMs in that positive units may be further restricted by syntactic disequations. Fermüller and Pichler [11] showed that the expressive power of DIGs corresponds to the one of so-called *contexts* used in the model evolution calculus [3] and that the ground atom test as well as the clause evaluation test and the equivalence test are decidable.

We extend the results of Fermüller and Pichler [11] for DIGs and ARMs to more expressive formulae with quantifier alternations using saturation-based techniques. We first enhance the non-equational part of our superposition calculus for fixed domains [14] with syntactic disequations (Section 3). The result is an ordered resolution calculus for fixed domains with syntactic disequations that is sound (Proposition 3.5) and complete (Theorem 3.9). Given an ARM representation N^*, we show in Section 6 that

$$\mathcal{I}_{N^*} \models \forall\vec{x}.\exists\vec{y}.\phi \text{ and } \mathcal{I}_{N^*} \models \exists\vec{x}.\forall\vec{y}.\phi$$

are both decidable, where ϕ is an arbitrary quantifier-free formula (Theorem 6.9). For more expressive DIG representations N^*, we show among other results that

$$\mathcal{I}_{N^*} \models \forall\vec{x}.\exists\vec{y}.C \text{ and } \mathcal{I}_{N^*} \models \exists\vec{x}.\forall\vec{y}.C'$$

are decidable for any clause C, and for any clause C' in which no predicate occurs both positively and negatively (Theorem 6.8). In order to cope with existential quantifiers in a minimal model semantics, we do not Skolemize but treat existential quantifiers by additional constraints.

This article is an extended version of [15].

2 Preliminaries

We build on the notions of [1, 19] and shortly recall here the most important concepts as well as the specific extensions needed for the new calculus COR presented in Section 3.

Terms and Clauses

Let $\Sigma = (\mathcal{P}, \mathcal{F})$ be a *signature* consisting of a finite set $\mathcal{P}$ of predicate symbols of fixed arity and a finite set $\mathcal{F}$ of function symbols of fixed arity, and let $X \cup V$ be an infinite set of variables such that X, V and $\mathcal{F}$ are disjoint and V is finite. Elements of X are called *universal variables* and denoted as x, y, z, and elements of V are called *existential variables* and denoted as v, possibly indexed.

We denote by $\mathcal{T}(\mathcal{F}, X')$ the set of all *terms* over $\mathcal{F}$ and $X' \subseteq X \cup V$ and by $\mathcal{T}(\mathcal{F})$ the set of all *ground terms* over $\mathcal{F}$. Throughout this article, we assume that $\mathcal{T}(\mathcal{F})$ is non-empty.

We will define equations and clauses in terms of multisets. A multiset over a set S is a function $M\colon S \to \mathbb{N}$. We use a list-like notation to describe multisets, e.g. A, A, A denotes the multiset M where $M(A) = 3$ and $M(B) = 0$ for all $B \neq A$ in S. If M and M' are two multisets, we denote the union of M and M' by M, M'.

An equation $s \simeq t$ is a multiset of two terms s, t; a disequation $s \not\simeq t$ is a multiset of two singleton sets of terms $\{s\}, \{t\}$. A multiset $s_1 \simeq t_1, \ldots, s_n \simeq t_n$ of equations is often written as $\vec{s} \simeq \vec{t}$. An *atom* over Σ is an expression of the form $P(t_1, \ldots, t_n)$, where $P \in \mathcal{P}$ is a predicate symbol of arity n and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, X)$ are terms. To improve readability, atoms $P(t_1, \ldots, t_n)$ will often be denoted as $P(\vec{t})$.

A *clause* is a pair of multisets of atoms, written $\Gamma \to \Delta$, interpreted as the conjunction of all atoms in the *antecedent* Γ implying the disjunction of all atoms in the *succedent* Δ. A clause is *Horn* if Δ contains at most one atom; it is a *unit* if the whole clause contains exactly one atom. The *empty* *clause* is denoted by $\Box$.

Constrained Clauses

A *constraint* α *over* $\Sigma = (\mathcal{P}, \mathcal{F})$ *(and* V*)* is a multiset of equations $s \simeq t$ and disequations $t \not\simeq t'$ where $s \in V \cup \mathcal{T}(\mathcal{F}, X)$ and $t, t' \in \mathcal{T}(\mathcal{F}, X)$ and every existential variable occurs at most once in α. We denote the multiset of equations in α by $\alpha^{\simeq}$ and the multiset of disequations by $\alpha^{\not\simeq}$. We call $\alpha^{\simeq}$ the *positive part* of α and say that α is *positive* if $\alpha = \alpha^{\simeq}$. A constraint is *ground* if it does not contain any universal variables.

Let $V = \{v_1, \ldots, v_n\}$ with $v_i \neq v_j$ for $i \neq j$. A *constrained clause* $\alpha \,\|\, C$ *over* Σ *(and* V*)* consists of a constraint α and a clause C over Σ, such that $\alpha^{\simeq} = v_1 \simeq t_1, \ldots, v_n \simeq t_n$. The set of all variables occurring in a constrained clause $\alpha \,\|\, C$ is denoted by $\mathrm{vars}(\alpha \,\|\, C)$. A constrained clause $\alpha \,\|\, C$ is called *ground* if it does not contain any universal variables, i.e. if $\mathrm{vars}(\alpha \,\|\, C) \subseteq V$ (and hence $\mathrm{vars}(\alpha \,\|\, C) = V$). We ignore linear constraint equations the variables of which do not occur elsewhere in the constrained clause, i.e. we abbreviate $(\alpha^{\simeq}, \beta) \,\|\, C$ as $\beta \,\|\, C$ if $\mathrm{vars}(\alpha^{\simeq}) \cap (\mathrm{vars}(\beta) \cup \mathrm{vars}(C)) = \emptyset$ and no variable appears twice in $\alpha^{\simeq}$. Such a constrained clause is called *unconstrained* if β is empty. We regard clauses as a special case of constrained clauses by identifying a clause C with $\,\|\, C$.

Substitutions

A *substitution* σ is a map from $X \cup V$ to $\mathcal{T}(\mathcal{F}, X)$ that acts as the identity map on all but a finite number of variables. We (non-uniquely) write $\sigma : Y \to Z$ if σ maps every variable in $Y \subseteq X \cup V$ to a term in $Z \subseteq \mathcal{T}(\mathcal{F}, X)$ and σ is the identity map on $(X \cup V) \setminus Y$. A substitution is identified with its extensions to terms, equations, atoms, and constrained clauses, where it is applied to both constraint and clausal part.

The most general unifier of two terms s, t or two atoms A, B is denoted by $\mathrm{mgu}(s, t)$ or $\mathrm{mgu}(A, B)$, respectively. If α_1 and α_2 are positive constraints of the form $\alpha_1 = v_1 \simeq s_1, \ldots, v_n \simeq s_n$ and $\alpha_2 = v_1 \simeq t_1, \ldots, v_n \simeq t_n$, then we write $\mathrm{mgu}(\alpha_1, \alpha_2)$ for the most general simultaneous unifier of $(s_1, t_1), \ldots, (s_n, t_n)$.

Orderings

Any ordering $\prec$ on atoms can be extended to unconstrained clauses in the following way. We consider clauses as multisets of occurrences of atoms. The occurrence of an atom A in the antecedent is identified with the multiset $\{A, A\}$; the occurrence of an atom A in the succedent is identified with the multiset $\{A\}$. Now we lift $\prec$ to atom occurrences as its multiset extension, and to clauses as the multiset extension of this ordering on atom occurrences.

An occurrence of an atom A is *maximal* in a clause C if there is no occurrence of an atom in C that is strictly greater with respect to $\prec$ than the occurrence of A. It is *strictly maximal* in C if there is no other occurrence of an atom in C that is greater than or equal to the occurrence of A with respect to $\prec$. Constrained clauses are ordered by their clausal part, i.e. $\alpha \,\|\, C \prec \beta \,\|\, D$ iff $C \prec D$.[1]

Throughout this paper, we will assume a well-founded reduction ordering $\prec$ on atoms over Σ that is total on ground atoms.

Herbrand Interpretations

A *Herbrand interpretation* over the signature Σ is a set of atoms over Σ. We recall from [1] the construction of the special Herbrand interpretation $\mathcal{I}_N$ derived from an unconstrained (and non-equational) clause set N. If N is consistent and saturated with respect to a complete inference system (possibly using a literal selection function), then $\mathcal{I}_N$ is a minimal model of N with respect to set inclusion. Let $\prec$ be a well-founded reduction ordering that is total on ground terms. We use induction on the clause ordering $\prec$ to define sets of atoms $\mathrm{Prod}(C), R(C)$ for ground clauses C over Σ. Let $\mathrm{Prod}(C) = \{A\}$ (and we say that C *produces* A), if $C = \Gamma \to \Delta, A$ is a ground instance of a clause $C' \in N$ such that (i) no literal in C' is selected, (ii) A is a strictly maximal occurrence of an atom in C, (iii) A is not an element of $R(C)$, (iv) $\Gamma \subseteq R(C)$, and (v) $\Delta \cap R(C) = \emptyset$. Otherwise $\mathrm{Prod}(C) = \emptyset$. In both cases, $R(C) = \bigcup_{C \succ C'} \mathrm{Prod}(C')$. Finally, we define the interpretation $\mathcal{I}_N = \bigcup_C \mathrm{Prod}(C)$ as the set of all produced atoms. We extend this construction of $\mathcal{I}_N$ to constrained clauses in section 3.

Constrained Clause Sets and Their Models

We interpret constraints as a conjunction and $\simeq$ and $\not\simeq$ as syntactic equality and disequality, respectively: We identify a constraint α over $\Sigma = (\mathcal{P}, \mathcal{F})$ with the formula $\phi_\alpha = \forall \vec{x}.(\bigwedge_{t \simeq t' \in \alpha} t{=}t' \wedge \bigwedge_{t \not\simeq t' \in \alpha} t{\neq}t')$, where $\vec{x}$ are the universal variables in α. We call α *satisfiable* if there is a Herbrand interpretation $\mathcal{M}$ over Σ in which ϕ_α is satisfiable, and write $\mathcal{M} \models \alpha$ iff $\mathcal{M} \models \phi_\alpha$. A ground constraint α that does not contain any variables is both satisfiable

[1] This ordering on constrained clauses differs from the one in [14]. Atoms there are equational, requiring superposition into constraints and that constrained clauses must also be ordered by their constraints. Constrained clauses here do not contain equational atoms but only syntactic (dis-)equations, so this is not necessary.

and valid (independent of the model $\mathcal{M}$) iff ϕ_α contains only equations $t{=}t$ and disequations $t{\neq}t'$ where t and t' are syntactically distinct.

Given a constrained clause set N, a Herbrand interpretation $\mathcal{M}$ over $\Sigma = (\mathcal{P}, \mathcal{F})$ is a *model* of N, written $\mathcal{M} \models N$, if and only if there is a substitution $\sigma : V \to \mathcal{T}(\mathcal{F})$ such that for every constrained clause $\alpha \,\|\, C \in N$ and every substitution $\tau : \text{vars}(\alpha \,\|\, C) \setminus V \to \mathcal{T}(\mathcal{F})$, that $\mathcal{M} \models \alpha\sigma\tau$ implies $\mathcal{M} \models C\tau$.[2] If N is finite, this is equivalent to the formula $\exists\vec{v}. \bigwedge_{\alpha \,\|\, C \in N} \forall\vec{x}.\alpha \to C$ being true in $\mathcal{M}$, where $\vec{x}$ are the universal variables in $\alpha \,\|\, C$ and $\simeq$ is interpreted as syntactic equality.[3] N is *Herbrand-satisfiable over* Σ if it has a Herbrand model over Σ.

Let M and N be two (constrained or unconstrained) clause sets. We write $N \models_\Sigma M$ if each Herbrand model of N over Σ is also a model of M, and we write $N \models_{\mathit{Ind}} M$ if $\mathcal{I}_N \models N$ and $\mathcal{I}_N \models M$.

Inferences, Redundancy and Derivations

An *inference rule* is a relation on constrained clauses. Its elements are called *inferences* and written as

$$\frac{\alpha_1 \,\|\, C_1 \;\; \ldots \;\; \alpha_k \,\|\, C_k}{\alpha \,\|\, C} \quad .$$

The constrained clauses $\alpha_1 \,\|\, C_1, \ldots, \alpha_k \,\|\, C_k$ are called the *premises* and $\alpha \,\|\, C$ the *conclusion* of the inference. An *inference calculus* is a set of inference rules.

A ground constrained clause $\alpha \,\|\, C$ is called *redundant* with respect to a set N of constrained clauses if α is unsatisfiable or if there are ground instances $\alpha_1 \,\|\, C_1, \ldots, \alpha_k \,\|\, C_k$ of constrained clauses in N with satisfiable constraints and the common positive constraint part $\alpha_1^{\simeq} = \ldots = \alpha_k^{\simeq} = \alpha^{\simeq}$ such that $\alpha_i \,\|\, C_i \prec \alpha \,\|\, C$ for all i and $C_1, \ldots, C_k \models C$.[4] A non-ground constrained clause is redundant if all its ground instances are redundant.

Given an inference system, a ground inference with conclusion $\alpha \,\|\, C$ is *redundant* with respect to N if some premise is redundant, or if there are ground instances $\alpha_1 \,\|\, C_1, \ldots, \alpha_k \,\|\, C_k$ of constrained clauses in N with satisfiable constraints and the common positive constraint part $\alpha_1^{\simeq} = \ldots = \alpha_k^{\simeq} = \alpha^{\simeq}$ such

[2] When considering constrained clauses, the usual definition of the semantics of a clause $\alpha \,\|\, C$ (where all variables are universally quantified) in the literature is simply the set of all ground instances $C\sigma$ such that σ is a solution of α (cf. [2, 18]). This definition does not meet our needs because we have existentially quantified variables, and these interconnect all clauses in a given constrained clause set.

[3] Finiteness of N is only required to ensure that the formula is also finite.

[4] Note that $\models$ and $\models_\Sigma$ agree on ground clauses.

that all $\alpha_i \,\|\, C_i$ are smaller than the maximal premise of the inference and $C_1, \ldots, C_k \models C$. A non-ground inference is redundant if all its ground instances are redundant. A constrained clause set N is *saturated* with respect to the inference system if each inference with premises in N is redundant with respect to N.

A *derivation* is a finite or infinite sequence $N_0, N_1, \ldots$ of constrained clause sets such that for each i, either (i) there is an inference with premises in N_i and conclusion $\alpha \,\|\, C$ such that $N_{i+1} = N_i \cup \{\alpha \,\|\, C\}$, or (ii) there is a clause $\alpha \,\|\, C \in N_i$ that is redundant with respect to N_i and $N_{i+1} = N_i \setminus \{\alpha \,\|\, C\}$. A derivation $N_0, N_1, \ldots$ is *fair* if every inference with premises in the constrained clause set $N_\infty = \bigcup_j \bigcap_{k \geq j} N_k$ is redundant with respect to $\bigcup_j N_j$.

3 A Constrained Ordered Resolution Calculus

In [14], we introduced a superposition-based calculus SFD to address the problem whether $N \models_\Sigma \forall\vec{x}.\exists\vec{y}.\phi$, where N is a set of unconstrained clauses and ϕ is a quantifier-free formula over Σ. There, both N and ϕ may contain equational atoms. The basic idea is to express the negation $\exists\vec{x}.\forall\vec{y}.\neg\phi$ of the query as a constrained clause set N' where all constraints are positive and the constraint part enables a special treatment of the existential variables without Skolemization. For example, $\exists x.\forall y.\neg P(x,y)$ corresponds to the constrained clause set $N' = \{v{\simeq}x \parallel P(x,y) \rightarrow\}$ with the unique existential variable v. If the saturation of $N \cup N'$ terminates, it is decidable whether $N \cup N'$ has a Herbrand model over Σ, i.e. whether $N \models_\Sigma \forall\vec{x}.\exists\vec{y}.\phi$. We recall this algorithm in Section 3.1.

In general, derivations with respect to SFD do not terminate. This is especially due to the two rules Constraint Superposition and Equality Elimination that will often produce clauses with ever increasing constraints. When only predicative atoms are present, several of the rules of SFD are not applicable, among them the two just mentioned. The remaining rules can be adapted to constraints containing both equations and disequations. The resulting calculus COR is presented in Section 3.2, along with proofs of its soundness and completeness with respect to Herbrand satisfiability.

3.1 The Fixed Domain Calculus SFD

The original superposition calculus for fixed domain reasoning SFD as presented in [14] works on clauses of equational atoms, but it only supports positive constraints. We use the symbol $\approx$ in these constrained clauses to distinguish between syntactic equality ($\simeq$) in the constraints and semantic equality ($\approx$) in the clausal parts. Predicates are encoded as equations as usual, i.e. $P(\vec{t})$ is encoded as $f_P(\vec{t})\approx$true, where true is a new and minimal constant symbol. If $\alpha_1 = \vec{v}\simeq\vec{s}$ and $\alpha_2 = \vec{v}\simeq\vec{t}$ are two positive constraints, then we write $\alpha_1\approx\alpha_2$ for the multiset $\vec{s}\approx\vec{t}$ of equations and we write $\text{mgu}(\alpha_1, \alpha_2)$ for the most general simultaneous unifier of $(s_1, t_1), \ldots, (s_n, t_n)$.

The rules of SFD are defined with respect to a *selection function* that assigns to each clause C a possibly empty set of atom occurrences in its antecedent. Such occurrences are called *selected*, and every inference with C has to use a selected atom of C. The calculus contains the following inference rules:

Equality Resolution:

$$\frac{\alpha \parallel \Gamma, s\approx t \to \Delta}{(\alpha \parallel \Gamma \to \Delta)\sigma}$$

where (i) $\sigma = \text{mgu}(s, t)$ and (ii) $(s\approx t)\sigma$ is maximal in $(\Gamma, s\approx t \to \Delta)\sigma$.

Equality Factoring:

$$\frac{\alpha \parallel \Gamma \to \Delta, s\approx t, s'\approx t'}{(\alpha \parallel \Gamma, t\approx t' \to \Delta, s'\approx t')\sigma}$$

where (i) $\sigma = \text{mgu}(s, s')$, (ii) $(s\approx t)\sigma$ is maximal in $(\Gamma \to \Delta, s\approx t, s'\approx t')\sigma$, and (iii) $t\sigma \not\succeq s\sigma$

Superposition, Right:

$$\frac{\alpha_1 \parallel \Gamma_1 \to \Delta_1, l\approx r \quad \alpha_2 \parallel \Gamma_2 \to \Delta_2, s[l']_p\approx t}{(\alpha_1 \parallel \Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2, s[r]_p\approx t)\sigma_1\sigma_2}$$

where (i) $\sigma_1 = \text{mgu}(l, l')$, $\sigma_2 = \text{mgu}(\alpha_1\sigma_1, \alpha_2\sigma_1)$, (ii) $(l\approx r)\sigma_1\sigma_2$ is strictly maximal in $(\Gamma_1 \to \Delta_1, l\approx r)\sigma_1\sigma_2$ and $(s\approx t)\sigma_1\sigma_2$ is strictly maximal in $(\Gamma_2 \to \Delta_2, s\approx t)\sigma_1\sigma_2$, (iii) $r\sigma_1\sigma_2 \not\succeq l\sigma_1\sigma_2$ and $t\sigma_1\sigma_2 \not\succeq s\sigma_1\sigma_2$, and (iv) l' is not a variable.

Superposition, Left:

$$\frac{\alpha_1 \parallel \Gamma_1 \to \Delta_1, l\approx r \quad \alpha_2 \parallel \Gamma_2, s[l']_p\approx t \to \Delta_2}{(\alpha_1 \parallel \Gamma_1, \Gamma_2, s[r]_p\approx t \to \Delta_1, \Delta_2)\sigma_1\sigma_2}$$

where (i) $\sigma_1 = \text{mgu}(l, l')$, $\sigma_2 = \text{mgu}(\alpha_1\sigma_1, \alpha_2\sigma_1)$, (ii) $(l\approx r)\sigma_1\sigma_2$ is strictly maximal in $(\Gamma_1 \to \Delta_1, l\approx r)\sigma_1\sigma_2$, $(s\approx t)\sigma_1\sigma_2$ is maximal in $(\Gamma_2 \to \Delta_2, s\approx t)\sigma_1\sigma_2$, (iii) $r\sigma_1\sigma_2 \not\succeq l\sigma_1\sigma_2$ and $t\sigma_1\sigma_2 \not\succeq s\sigma_1\sigma_2$, and (iv) l' is not a variable.

Constraint Superposition:

$$\frac{\alpha_1 \,\|\, \Gamma_1 \to \Delta_1, l{\approx}r \quad \alpha_2[l'] \,\|\, \Gamma_2 \to \Delta_2}{(\alpha_2[r] \,\|\, \alpha_1{\approx}\alpha_2[r], \Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2)\sigma}$$

where (i) $\sigma = \mathrm{mgu}(l, l')$, (ii) $(l{\approx}r)\sigma$ is strictly maximal in $(\Gamma_1 \to \Delta_1, l{\approx}r)\sigma$, (iii) $r\sigma \not\succeq l\sigma$, and (iv) l' is not a variable.

Equality Elimination:

$$\frac{\alpha_1 \,\|\, \Gamma \to \Delta, l{\approx}r \quad \alpha_2[r'] \,\|\, \Box}{(\alpha_1 \,\|\, \Gamma \to \Delta)\sigma_1\sigma_2}$$

where (i) $\sigma_1 = \mathrm{mgu}(r, r')$, $\sigma_2 = \mathrm{mgu}(\alpha_1\sigma_1, \alpha_2[l]\sigma_1)$, (ii) $(l{\approx}r)\sigma_1\sigma_2$ is strictly maximal in $(\Gamma \to \Delta, l{\approx}r)\sigma_1\sigma_2$, (iii) $r\sigma_1\sigma_2 \not\succeq l\sigma_1\sigma_2$, and (iv) r' is not a variable.

This calculus is both sound and complete for fixed domain reasoning, which, in the case of constrained Horn clauses, coincides with inductive reasoning:

Proposition 3.1 (Soundness [14, Proposition 3])**.** *Let $\alpha \,\|\, C$ be the conclusion of an inference with premises in a constrained clause set N.*

Then $N \models_{Ind} N \cup \{\alpha \,\|\, C\}$.

Theorem 3.2 (Completeness [14, Theorem 1])**.** *Let N be a set of constrained Horn clauses over a signature Σ that is saturated with respect to the calculus SFD.*

Then N does not have a Herbrand model over Σ iff every positive ground constraint over Σ is an instance of a constraint α such that $\alpha \,\|\, \Box \in N$.

3.2 The Calculus COR

In our current setting with constrained clauses that contain only predicative atoms, only two rules from the original calculus remain. Extending these rules to clauses with constraints containing both equations and disequations yields the following calculus.

Definition 3.3. The *constrained ordered resolution calculus COR* consists of the following two inference rules:

Ordered Resolution:

$$\frac{\alpha_1 \,\|\, \Gamma_1 \rightarrow \Delta_1, A_1 \quad \alpha_2 \,\|\, \Gamma_2, A_2 \rightarrow \Delta_2}{(\alpha_1, \alpha_2^{\not\simeq} \,\|\, \Gamma_1, \Gamma_2 \rightarrow \Delta_1, \Delta_2)\sigma_1\sigma_2}$$

where (i) $\sigma_1 = \mathrm{mgu}(A_1, A_2)$ and $\sigma_2 = \mathrm{mgu}(\alpha_1^{\simeq}\sigma_1, \alpha_2^{\simeq}\sigma_1)$,(ii) no atom is selected in $\Gamma_1 \rightarrow \Delta_1, A_1$ and $A_1\sigma_1\sigma_2$ strictly maximal in $(\Gamma_1 \rightarrow \Delta_1, A_1)\sigma_1\sigma_2$, and (iii) either A_2 is selected in $\Gamma_2, A_2 \rightarrow \Delta_2$, or no atom occurrence is selected in $\Gamma_2, A_2 \rightarrow \Delta_2$ and $A_2\sigma_1\sigma_2$ is maximal in $(\Gamma_2, A_2 \rightarrow \Delta_2)\sigma_1\sigma_2$.

Ordered Factoring:

$$\frac{\alpha \,\|\, \Gamma \rightarrow \Delta, A, A'}{(\alpha \,\|\, \Gamma \rightarrow \Delta, A)\sigma}$$

where (i) $\sigma = \mathrm{mgu}(A, A')$ and (ii) no occurrence is selected in $\Gamma \rightarrow \Delta, A, A'$ and $A\sigma$ is maximal in $(\Gamma \rightarrow \Delta, A, A')\sigma$.

We unify $\alpha_1^{\simeq}$ and $\alpha_2^{\simeq}$ in the Ordered Resolution rule for efficiency reasons: A natural alternative would be to consider constraints with multiple occurrences of the same existential variable and let the constraint of the conclusion be $(\alpha_1, \alpha_2)\sigma_1\sigma_2$ instead of $(\alpha_1, \alpha_2^{\not\simeq})\sigma_1\sigma_2$. But since equations are syntactic, if $\alpha_1^{\simeq}$ and $\alpha_2^{\simeq}$ are not unifiable then any variable-free instance of α_1, α_2 is unsatisfiable anyway, so this would not add expressiveness to the calculus.

Further refinements are of course possible, e.g. by restricting Ordered Resolution to instances where $(\alpha_1^{\not\simeq}, \alpha_2^{\not\simeq})\sigma_1\sigma_2$ is satisfiable.

We will now show that we can decide whether a finite constrained clause set N that is saturated with respect to COR has a Herbrand model over Σ and, if so, how to construct such a model. As constrained clauses are an extension of unconstrained clauses, the construction of a Herbrand model of N is strongly related to the one from [1] for unconstrained clause sets presented in Section 2. The main difference is that we now have to account for constraints before starting the construction. To define a Herbrand interpretation $\mathcal{I}_N$ of a set N of constrained clauses over Σ, we proceed in two steps: First we identify a constraint α_N that is a potential witness of the satisfiability of N, corresponding to an instantiation of the existential variables. Then we apply this instantiation and build an interpretation for the now purely universal clause set in the usual superposition-style way.

(1) Let $V = \{v_1, \ldots, v_n\}$ and let $A_N = \{\alpha \mid (\alpha \,\|\, \Box) \in N\}$ be the set of all constraints of constrained clauses in N with empty clausal part. A_N is

covering (for Σ) if for every positive ground constraint $\beta = \vec{v}\simeq\vec{t}$ over Σ there is a satisfiable ground instance $\alpha\tau$ of a constraint $\alpha \in A_N$ such that $\beta = \alpha^{\simeq}\tau$.

We define the constraint α_N as follows: If A_N is not covering, let $\alpha_N = \vec{v}\simeq\vec{t}$ be a positive ground constraint that is not equal to the equational part of any satisfiable ground instance of a constraint in A_N.[1] If A_N is covering, let α_N be an arbitrary ground constraint. We will show that N is Herbrand-unsatisfiable over Σ in this case.

(2) $\mathcal{I}_N$ is defined as the Herbrand interpretation $\mathcal{I}_{N'}$ associated to the (unconstrained) ground clause set

$$N' = \{C\tau \mid (\alpha \,\|\, C) \in N \text{ and } \tau : \mathrm{vars}(\alpha \,\|\, C) \setminus V \to \mathcal{T}(\mathcal{F}) \\ \text{and } \alpha_N = \alpha^{\simeq}\tau \text{ and } \alpha\tau \text{ is satisfiable}\} .$$

Example 3.4. As an example consider the signature $\Sigma = (\emptyset, \{0, s\})$, where 0 is a constant and s is unary, a single existential variable v and the two constrained clause sets $M = \{v\simeq 0 \,\|\, \Box,\ v\simeq s(0) \,\|\, \Box,\ v\simeq s(s(0)) \,\|\, \Box,\ \ldots\}$ and $N = \{v\simeq s(x), x\not\simeq 0 \,\|\, \Box\}$. Then A_M is covering but A_N is not, and we may choose either $\{v\simeq 0\}$ or $\{v\simeq s(0)\}$ for α_N.

One easily sees that the interpretation $\mathcal{I}_N$ is independent of α_N if all clauses in N are unconstrained. If moreover $\mathcal{I}_N \models N$, then $\mathcal{I}_N$ is a minimal model of N with respect to set inclusion. We call $\mathcal{I}_N$ *the minimal model* of N in this case.

While it is well known how the second step in the construction of $\mathcal{I}_N$ works, it is not obvious that one can decide whether A_N is covering and, if it is not, effectively compute some α_N. This is, however, possible for finite A_N: Let $\{x_1, \ldots, x_m\} \subseteq X$ be the set of universal variables appearing in A_N. By definition, A_N is covering if and only if the formula $\forall\vec{v}.\exists\vec{x}.\bigvee_{\alpha\in A_N}\alpha$ is true if and only if $\forall\vec{x}.\bigwedge_{\alpha\in A_N}\neg\alpha$ is unsatisfiable in $\mathcal{T}(\mathcal{F})$. Such so-called disunification problems have been studied among others by Comon and Lescanne [7], who gave a terminating algorithm that eliminates the universal quantifiers from this formula and transforms the initial problem into an equivalent solved form from which the set of solutions can easily be read off.

For saturated sets, the information contained in the constrained empty clauses is already sufficient to decide whether Herbrand models exist. Specifically, we will now show that Herbrand-satisfiability or unsatisfiability over Σ is invariant under the application of inferences in COR (Proposition 3.6)

[1] In contrast to [14], we do not have to impose any minimality requirements on α_N.

and that a saturated constrained clause set N has a Herbrand model over Σ (namely $\mathcal{I}_N$) if and only if A_N is not covering (Propositions 3.7 and 3.8).

Proposition 3.5 (Soundness)**.** *Let $\alpha \,\|\, C$ be the conclusion of a COR inference with premises in a set N of constrained clauses over Σ. Then $N \models_\Sigma N \cup \{\alpha \,\|\, C\}$.*

Proof. We have to show that every Herbrand model of N over $\Sigma = (\mathcal{P}, \mathcal{F})$ is also a Herbrand model of $N \cup \{\alpha \,\|\, C\}$. Let $\mathcal{M}$ be a Herbrand model of N over Σ. Then there is a substitution $\sigma : V \to \mathcal{T}(\mathcal{F})$ such that for every constrained clause $\alpha \,\|\, C \in N$ and every substitution $\tau : \text{vars}(\alpha \,\|\, C) \setminus V \to \mathcal{T}(\mathcal{F})$, if $\alpha\sigma\tau$ is satisfiable then $\mathcal{M} \models C\tau$.

Fix such a σ and consider an ordered resolution inference as follows:

$$\frac{\alpha_1 \,\|\, C_1 \quad \alpha_2 \,\|\, C_2}{(\alpha_1, \alpha_2^{\not\simeq} \,\|\, C)\tau}$$

Let $\tau' : \text{vars}((\alpha_1, \alpha_2^{\not\simeq} \,\|\, C)\tau) \setminus V \to \mathcal{T}(\mathcal{F})$. If $(\alpha_1, \alpha_2^{\not\simeq})\tau\sigma\tau'$ is satisfiable, then so are $\alpha_1\sigma\tau\tau'$ and $\alpha_2\sigma\tau\tau'$. Because $\mathcal{M}$ is a Herbrand model of N over Σ, this implies that $\mathcal{M} \models C_1\tau\tau'$ and $\mathcal{M} \models C_2\tau\tau'$. Because *unconstrained* ordered resolution is sound, we can conclude that $\mathcal{M} \models C\tau\tau'$.

The proof for an ordered factoring inference is similar. $\Diamond$

As usual, the fairness of a derivation can be ensured by systematically adding conclusions of non-redundant inferences, making these inferences redundant.

The following proposition relies on soundness, redundancy, and fairness rather than on a concrete inference system. Hence its proof is exactly as in the unconstrained case or in the case of the original fixed domain calculus (cf. [1, 14]):

Proposition 3.6 (Saturation)**.** *Let $N_0, N_1, \ldots$ be a fair COR derivation. Then the set $N_\infty = \bigcup_j \bigcap_{k \geq j} N_k$ is saturated. N_0 has a Herbrand model over Σ if and only if N_∞ does.*

Now we express the Herbrand-satisfiability of N^* over Σ in terms of the coverage of A_{N^*}.

Proposition 3.7. *Let N be a set of constrained clauses such that A_N is covering. Then N does not have any Herbrand model over Σ.*

Proof. Let $\mathcal{M}$ be a Herbrand model of N over $\Sigma = (\mathcal{P}, \mathcal{F})$. Then there is a substitution $\sigma_\mathcal{M} : V \to \mathcal{T}(\mathcal{F})$ such that for every constrained clause $\alpha \,\|\, C \in N$ and every substitution $\tau : \text{vars}(\alpha \,\|\, C) \setminus V \to \mathcal{T}(\mathcal{F})$, if $\alpha\sigma_\mathcal{M}\tau$ is satisfiable then $\mathcal{M} \models C\tau$. We show that the constraint $\alpha_\sigma = v_1 \simeq v_1\sigma_\mathcal{M}, \ldots, v_n \simeq v_n\sigma_\mathcal{M}$ is not covered by A_N.

It clearly holds that α_σ is satisfiable and $\mathcal{M} \models \alpha_\sigma\sigma$. If the constraint α_σ were of the form $\alpha_\sigma = \alpha\tau$ for a constrained clause $\alpha \,\|\, \square \in N$ and some substitution τ, it would thus follow that $\mathcal{M} \not\models \alpha \,\|\, \square$, which contradicts the fact that $\mathcal{M}$ is a model of N. $\Diamond$

Proposition 3.8 (Σ-Completeness for Saturated Clause Sets)**.** *Let N be a saturated set of constrained clauses over Σ such that A_N is not covering for Σ. Then $\mathcal{I}_N \models N$ for any admissible choice of α_N.*

Proof. Let $\alpha_N = v_1{\simeq}t_1, \ldots, v_n{\simeq}t_n$ and assume, contrary to the proposition, that N is not modeled by $\mathcal{I}_N$. Then there are $\alpha \,\|\, C \in N$ and $\sigma : \text{vars}(\alpha \,\|\, C) \to \mathcal{T}(\mathcal{F})$ such that $\mathcal{I}_N \not\models (\alpha \,\|\, C)\sigma$, This implies that $\sigma(v_i) = t_i$ for all i, $\alpha\sigma$ is satisfiable, and $\mathcal{I}_N \not\models C\sigma$. Let $C\sigma$ be minimal with these properties.

We will refute this minimality. We proceed by a case analysis of the position of selected or maximal literal occurrences in $C\sigma$.

- $C\sigma$ does not contain any literal at all, i.e. $C = \square$. Then the satisfiability of $\alpha\sigma$ contradicts the choice of α_N.

- $C = \Gamma, A \to \Delta$ and $A\sigma$ is selected or $A\sigma$ is maximal and no literal is selected in $C\sigma$. Since $\mathcal{I}_N \not\models C\sigma$, we know that $A\sigma \in \mathcal{I}_N$. The literal A must be produced by a ground instance $(\beta \,\|\, \Lambda \to \Pi, B)\sigma'$ of a constrained clause in N in which no literal is selected. Note that both ground constrained clauses $(\alpha \,\|\, C)\sigma$ and $(\beta \,\|\, \Lambda \to \Pi, B)\sigma'$ are not redundant with respect to N.

  Because $\alpha^{\simeq}\sigma = \beta^{\simeq}\sigma' = \alpha_N\sigma$ and because σ is a unifier of A and B, i.e. an instance of $\sigma_1 := \text{mgu}(A, B)$, there is an inference by ordered resolution as follows:

  $$\frac{\beta \,\|\, \Lambda \to \Pi, B \quad \alpha \,\|\, \Gamma, A \to \Delta}{(\alpha, \beta^{\not\simeq} \,\|\, \Lambda, \Gamma \to \Pi, \Delta)\sigma_1\sigma_2}\ \sigma_2 = \text{mgu}(\beta^{\simeq}\sigma_1, \alpha^{\simeq}\sigma_1)$$

  Looking at the ground instance $\delta \,\|\, D = (\alpha, \beta^{\not\simeq} \,\|\, \Lambda, \Gamma \to \Pi, \Delta)\sigma$ of the conclusion, we see that δ is satisfiable and $\mathcal{I}_N \not\models D$.

  On the other hand, as the inference is redundant, so is the constrained clause $\delta \,\|\, D$, i.e. D follows from ground instances $\delta \,\|\, C_i$ of constrained clauses of N all of which are smaller than $(\alpha \,\|\, C)\sigma$. Because of the minimality of $C\sigma$, all C_i hold in $\mathcal{I}_N$. So $\mathcal{I}_N \models D$, which contradicts $\mathcal{I}_N \not\models D$.

- $C = \Gamma \to \Delta, A$ and $A\sigma$ is strictly maximal in $C\sigma$. This is not possible, since then either $C\sigma$ or a smaller clause must have produced $A\sigma$, and hence $\mathcal{I}_N \models C\sigma$, which contradicts the choice of $C\sigma$.

- No literal in $C = \Gamma \to \Delta, A$ is selected and $A\sigma$ is maximal but not strictly maximal in $C\sigma$. Then $\Delta = \Delta', A'$ such that $A'\sigma = A\sigma$. So there is an inference by ordered factoring as follows:

  $$\frac{\alpha \,\|\, \Gamma \to \Delta', A, A'}{(\alpha \,\|\, \Gamma \to \Delta', A')\sigma_1} \ \sigma_1 = \mathrm{mgu}(s, s')$$

  As above, $\alpha\sigma$ is satisfiable and we can derive both $\mathcal{I}_N \models (\Gamma \to \Delta', A')\sigma$ and $N \not\models (\Gamma \to \Delta', A')\sigma$, which is a contradiction.

$\Diamond$

Combining Propositions 3.6, 3.7 and 3.8, we can conclude:

Theorem 3.9. *Let $N_0, N_1, \ldots$ be a fair COR derivation. Then the constrained clause set $N^* = \bigcup_j \bigcap_{k \geq j} N_k$ is saturated. Moreover, N_0 has a Herbrand model over Σ if and only if A_{N^*} is not covering.*

4 Predicate Completion

The calculus COR can be used to reason about all Herbrand models of a constrained clause set N over a signature Σ: Theorem 3.9 states that $N \models_\Sigma \phi$ is decidable when $\neg\phi$ can be represented as a set M of clauses and the saturation of $N \cup M$ using COR terminates. However, it is usually desirable to be able to reason about single models, e.g. the minimal model $\mathcal{I}_N$. The key to focusing on this model is *predicate completion*, an enrichment of N to a larger constrained clause set N' in such a way that N' does not have any Herbrand models over Σ other than $\mathcal{I}_N$.

We will now present the predicate completion approach by Comon and Nieuwenhuis [8] (Section 4.2). This approach relies on a quantifier elimination procedure that is presented in Section 4.1.

4.1 Quantifier Elimination

Quantifier elimination provides a means to transform a formula into a simple normal form for which satisfiability with respect to the free term algebra is easily decidable.

In this section, we present the rules of the quantifier elimination algorithm by Comon and Lescanne [7] as they are given, in a revised version, in [6]. To simplify the presentation, we omit applicability constraints that are only relevant for the termination of the rule system. In the rules, different designators stand for elements of restricted sets of variables or terms:

$x, x_i, \ldots$	free variables
$y, y_i, \ldots$	universally quantified variables
$v, v_i, \ldots$	existentially quantified variables
$z, z_i, \ldots$	free or existentially quantified variables
$w, w_i, \ldots$	any variables
$t, t_i, \ldots$	terms without any occurrence of a universally quantified variable
$s, s_i, u, u_i \ldots$	terms
$u[u'], \ldots$	the term u' is a subterm of the term u
$d, d_i, \ldots$	disjunctions of equations and disequations and predicative literals
$\phi, \phi_i, \psi, \psi_i, \ldots$	quantifier-free formulas

While it is unusual that three kinds of variables (free, universal, and existential) appear, this variety stems from the intended application of finding solutions, i.e. satisfying instantiations of the free variables, of a formula containing existential and universal quantifiers. Because of this, the free variables require a special treatment and cannot be regarded as bound by an additional quantifier on top of the formula.

The quantifier elimination rules are given below. They can be applied at any position of a formula. The capture-free substitution of a variable x by a term t in a formula ϕ is written as $\phi\{x \mapsto t\}$.

Replacement:

R1: $z \simeq t \wedge \phi[z] \leadsto z \simeq t \wedge \phi\{z \mapsto t\}$

R2: $z \not\simeq t \vee \phi[z] \leadsto z \not\simeq t \vee \phi\{z \mapsto t\}$

If $z \notin \text{vars}(t)$.

Universal quantifier elimination:

UE1: $\forall \vec{y}, y.\phi \leadsto \forall \vec{y}.\phi$

UE2: $\phi \wedge y \not\simeq u \leadsto \text{false}$

UE3: $y \not\simeq u \vee d \leadsto d\{y \mapsto u\}$

UE5: $\phi \wedge (w_1 \simeq u_1 \vee \ldots \vee w_n \simeq u_n \vee d) \leadsto \phi \wedge d$

If $y \notin \text{vars}(\phi)$ (for UE1), $y \notin \text{vars}(u)$ (for UE2-3) and each $w_i \simeq u_i$ contains at least one occurrence of a universally quantified variable, but d does not (for UE5). UE5 is only applicable if $\mathcal{T}(\mathcal{F})$ is infinite.

Merging:

M1: $z \simeq t \wedge z \simeq u \leadsto z \simeq t \wedge t \simeq u$

M2: $w \not\simeq s \vee w \not\simeq u \leadsto w \not\simeq s \vee s \not\simeq u$

M3: $z \simeq t \wedge z \not\simeq u \leadsto z \simeq t \wedge t \not\simeq u$

M4: $w \simeq s \vee w \not\simeq u \leadsto s \simeq u \vee w \not\simeq u$

M5: $z \simeq t \wedge (z \not\simeq u \vee \phi) \leadsto z \simeq t \wedge (t \not\simeq u \vee \phi)$

M6: $z \simeq t \wedge (z \simeq u \vee \phi) \leadsto z \simeq t \wedge (t \simeq u \vee \phi)$

The merging rules are not necessary for termination or completeness.

Existential quantifier elimination:

$$\text{EE1:} \quad \exists v.\phi \rightsquigarrow \phi$$
$$\text{EE2:} \quad \exists \vec{v}, v.(v \simeq t \wedge \phi) \rightsquigarrow \exists \vec{v}.\phi$$
$$\text{EE3:} \quad \exists \vec{v}.(\psi \wedge \phi) \rightsquigarrow \exists \vec{v}.\phi$$

If $v \notin \text{vars}(\phi, t)$ (for EE1-2) and $\psi = (d_1 \vee z_1 \not\simeq t_1) \wedge \ldots \wedge (d_n \vee z_n \not\simeq t_n)$ and there exists $v \in \vec{v} \cap \text{vars}(z_1, t_1) \cap \text{vars}(z_n, t_n)$ which does not occur in ϕ and $\mathcal{T}(\mathcal{F})$ is infinite (for EE3).

Normalization:

Rules for the transformation into negation normal form and conjunctive normal form, propagation of truth and falsity, and Boolean simplification rules like $\phi \wedge \phi \rightarrow \phi$.

Disjunction lifting:

$$\text{DL:} \ \exists \vec{v}.\forall \vec{y}.(\phi \wedge (\phi_1 \vee \phi_2)) \rightsquigarrow (\exists \vec{v}.\forall \vec{y}.(\phi \wedge \phi_1)) \vee (\exists \vec{v}.\forall \vec{y}.(\phi \wedge \phi_2))$$

If $\text{vars}(\phi_1) \cap \vec{y} = \emptyset$ or $\text{vars}(\phi_2) \cap \vec{y} = \emptyset$.

Conflict:

$$\text{C1:} \ f(u_1, \ldots, u_m) \simeq g(u'_1, \ldots, u'_n) \rightsquigarrow \text{false}$$
$$\text{C2:} \ f(u_1, \ldots, u_m) \not\simeq g(u'_1, \ldots, u'_n) \rightsquigarrow \text{true}$$

If $f \neq g$.

Decomposition:

$$\text{D1:} \ f(u_1, \ldots, u_n) \simeq f(u'_1, \ldots, u'_n) \rightsquigarrow u_1 \simeq u'_1 \wedge \ldots \wedge u_n \simeq u'_n$$
$$\text{D2:} \ f(u_1, \ldots, u_n) \not\simeq f(u'_1, \ldots, u'_n) \rightsquigarrow u_1 \not\simeq u'_1 \vee \ldots \vee u_n \not\simeq u'_n$$

Occurrence check:

$$\text{O1:} \ s \simeq u[s] \rightsquigarrow \text{false}$$
$$\text{O2:} \ s \not\simeq u[s] \rightsquigarrow \text{true}$$

If $u \neq s$.

Explosion:

$$\text{E:} \ \exists \vec{v}.\forall \vec{y}.\phi \rightsquigarrow \bigvee_{f \in \mathcal{F}} \exists \vec{v}_1, \vec{v}.\forall y.\phi \wedge z \simeq f(\vec{v}_1)$$

If no other rule can be applied, the variables $\vec{v}_1$ are fresh and there exists in ϕ an equation $z \simeq u$ or disequation $z \not\simeq u$ where u contains an occurrence of a universally quantified variable.

By application of these rules, universal quantifiers can be eliminated from purely equational formulas of a certain shape:

Theorem 4.1 ([6, Theorem 9])**.** *Let $\psi = \bigvee_j (\exists \vec{w}_j.\forall \vec{y}_j.\phi_j)$ be a formula such that each ϕ_j is a quantifier-free formula over a finite signature $\Sigma = (\mathcal{P}, \mathcal{F})$.*

Then the non-deterministic application of the above rule set transforms ψ *in finitely many steps into a formula* ψ' *of the form*

$$\psi' = \bigvee_j (\exists \vec{w}_j . x_{j1} \simeq t_{j1} \wedge \ldots \wedge x_{jn_j} \simeq t_{jn_j} \wedge z_{j1} \not\simeq t'_{j1} \wedge \ldots \wedge z_{jm_j} \not\simeq t'_{jm_j})$$

where the x_{ji} *are variables occurring only once in each disjunct, and each* z_{jk} *is a variable that is not identical with* t'_{jk}*.*

Moreover, ψ *and* ψ' *are satisfied by the same assignments of terms in* $\mathcal{T}(\mathcal{F})$ *to the free variables.*

4.2 Predicate Completion

Predicate completion [5] is the process of extending unconstrained clause sets in such a way that all Herbrand models but the minimal one are excluded.

Comon and Nieuwenhuis [8] described a concrete predicate completion algorithm for unconstrained *Horn* clause sets. However, the interpretations associated with DIGs and ARMs that we will be interested in in Section 5 will be described as minimal models of usually non-Horn clause sets. In what follows, we generalize the predicate completion procedure by Comon and Nieuwenhuis to unconstrained clause sets that are not necessarily Horn. We then show that the extended algorithm computes a completion for so-called stepwise saturated unconstrained clause sets, which includes all unconstrained clause sets that will appear in this paper.

The predicate completion algorithm proceeds as follows:

(1) Let P be a predicate and let N_P be a finite set of unconstrained clauses over $\Sigma = (\mathcal{P}, \mathcal{F})$ all of which are of the form $\Gamma \rightarrow \Delta, P(\vec{t})$, where $P(\vec{t})$ is a strictly maximal literal occurrence. Combine all these clauses into a single formula $\forall \vec{x}.(\phi_P \rightarrow P(\vec{x}))$, where

$$\phi_P = \exists \vec{y}. \bigvee_{\Gamma \rightarrow \Delta, P(\vec{t}) \in N_P} (\vec{x} \simeq \vec{t} \wedge \bigwedge_{A \in \Gamma} A \wedge \bigwedge_{B \in \Delta} \neg B) ,$$

the y_i are the variables appearing in N_P, and the x_j are fresh variables.

(2) In the minimal model $\mathcal{I}_{N_P}$, the formula $\forall \vec{x}.(\phi_P \rightarrow P(\vec{x}))$ is equivalent to the formula $\forall \vec{x}.(\neg \phi_P \rightarrow \neg P(\vec{x}))$. If all variables appearing in $\Gamma \rightarrow \Delta, P(\vec{t})$ also appear in $P(\vec{t})$, then $\neg \phi_P$ can be transformed using quantifier elimination into an equivalent formula ψ that does not contain any universal quantifiers. This quantifier elimination procedure is the same that we used in Section 3 to decide the coverage of constraint sets.

(3) The formula ψ can in turn be written as a set N'_P of constrained clauses. The union $N_P \cup N'_P$ is a *completion* of N_P.

(4) If N is the union of several sets as in (1) defining different predicates, then the completion N' of N is the union of N and all N'_P, $P \in \mathcal{P}$, where N_P is the set of all clauses in N having a strictly maximal literal $P(\vec{t})$.

Comon and Nieuwenhuis showed that the minimal model of a Herbrand-satisfiable unconstrained Horn clause set is also a model of its completion:

Lemma 4.2 ([8, Lemma 47])**.** *Let N be a satisfiable unconstrained Horn clause set over Σ and let N' be a completion of N. Then $\mathcal{I}_N$ is the unique Herbrand model of N' over Σ.*

This result extends also to some sets of non-Horn clauses:

Definition 4.3. Let N be a set of unconstrained clauses. A predicate P is called *of level* 1 *in* N if every clause $C \in N$ containing a positive literal $P(\vec{t})$ is of the form $C = \ \rightarrow P(\vec{t})$. The predicate is called *of level* $n+1$ *in* N if every clause in N containing a positive literal $P(\vec{t})$ is of the form $\Gamma \rightarrow \Delta, P(\vec{t})$, where $P(\vec{t})$ is strictly maximal, every predicate Q occurring in Γ and Δ is of level $\leq n$ in N, and n is minimal with this property.

Definition 4.4. Let N be a set of unconstrained clauses. Let N_n be the restriction of N to clauses that contain only predicates of level $\leq n$ in N. Then N is called *stepwise saturated* if (i) N_n is saturated for all $n \geq 1$ and (ii) every predicate has a finite level in N, i.e. $N = N_m$ for some $m \geq 1$.

Proposition 4.5. *Let N be a satisfiable and stepwise saturated set of unconstrained clauses and let $\mathcal{M}$ be a model of the completion N' of N.*

Then particular $\mathcal{M} = \mathcal{I}_N$.

Proof. For $n \in \mathbb{N}$, let N_n be the restriction of N to clauses that contain only predicates of level $\leq n$ and let $\mathcal{M}_n$ and $\mathcal{I}_n$ be the restrictions of $\mathcal{M}$ and $\mathcal{I}_N$ to atoms $P(\vec{t})$ where P is of level $\leq n$ in N. Note that due to the stepwise saturation of N, it holds that $\mathcal{I}_n = \mathcal{I}_{N_n}$.

We inductively show that $\mathcal{M}_n = \mathcal{I}_n$ for all $n \geq 1$.

For $n = 1$, the proposition follows from Lemma 4.2.

For $n > 1$, assume that $\mathcal{M}_{n-1} = \mathcal{I}_{n-1}$. We separately prove the two inclusions $\mathcal{M}_n \subseteq \mathcal{I}_n$ and $\mathcal{M}_n \supseteq \mathcal{I}_n$.

Let $P(\vec{t}) \in \mathcal{I}_n$ be produced by a ground instance $\Gamma \rightarrow \Delta, P(\vec{t})$ of a clause in N_n. Then $\Gamma \subseteq \mathcal{I}_n$ and $\Delta \cap \mathcal{I}_n = \emptyset$. Because P is of level $\leq n$ and so all predicates appearing in Γ and Δ are of level $< n$ in N, it also holds that

$\Gamma \subseteq \mathcal{I}_{n-1} = \mathcal{M}_{n-1} \subseteq \mathcal{M}_n$ and $\Delta \cap \mathcal{M}_n = \Delta \cap \mathcal{M}_{n-1} = \Delta \cap \mathcal{I}_{n-1} = \emptyset$. Since $\Gamma \to \Delta, P(\vec{t})$ must also hold in $\mathcal{M}_n$, this implies $P(\vec{t}) \subseteq \mathcal{M}_n$.

On the other hand, let $P(\vec{t}) \in \mathcal{M}_n$. All clauses in N' and hence the formula $\forall \vec{x}.\neg\phi_P \to \neg P(\vec{x})$ is valid in $\mathcal{M}$, i.e. $\mathcal{M} \models \phi_P[\vec{x} \mapsto \vec{t}]$. All predicates in ϕ_P are of level $< n$, so $\mathcal{M}_{n-1} \models \phi_P[\vec{x} \mapsto \vec{t}]$. Because $\mathcal{M}_{n-1} = \mathcal{I}_{n-1}$ and because $\mathcal{I}_{n-1}$ and $\mathcal{I}_n$ agree on predicates of level $< n$, if follows that $\mathcal{I}_n \models \phi_P[\vec{x} \mapsto \vec{t}]$. Because $\mathcal{I}_N \models N$, the formula $\forall \vec{x}.\phi_P \to P(\vec{x})$ is valid in $\mathcal{I}_N$. The formula contains only predicates of level $\leq n$, so it is also valid in $\mathcal{I}_n$. Together with $\mathcal{I}_n \models \phi_P[\vec{x} \mapsto \vec{t}]$, this implies $\mathcal{I}_n \models P(\vec{t})$. $\Diamond$

We will see an example of the completion of a stepwise saturated clause set in Section 5.3 (Example 5.10).

5 Clausal Representations of Disjunctions of Implicit Generalizations

5.1 Disjunctions of Implicit Generalizations

In Section 3, we showed how saturated sets of constrained clauses can be regarded as (implicitly) representing certain Herbrand models. Other representations of Herbrand models include sets of non-ground atoms or the more flexible so-called disjunctions of implicit generalizations of Lassez and Marriott [17]. We show how both types of representation can be seen as special cases of the representation by saturated constrained clause sets.

Based on this view, we reprove that the equivalence of any given pair of representations by disjunctions of implicit generalizations is decidable, and we extend the known results on the decidability of clause and formula entailment (cf. [11]). To do so, we translate a query $\mathcal{M} \models \phi$ over a signature Σ into a constrained clause set that is Herbrand-unsatisfiable over Σ iff $\mathcal{M} \models \phi$ holds. The Herbrand-unsatisfiability can then be decided using the calculus COR.

Definition 5.1. An *implicit generalization* G over Σ is an expression of the form $G = A/\{A_1, \ldots, A_n\}$, where $A, A_1, \ldots, A_n$ are atoms over Σ. A finite set D of implicit generalizations over Σ is called a *DIG (disjunction of implicit generalizations)*. A DIG D is an *atomic representation of a term model (ARM)*, if all implicit generalizations in D are of the form $A/\{\}$.

The Herbrand model $\mathcal{M}(\{A/\{A_1, \ldots, A_n\}\})$ represented by a DIG consisting of a single implicit generalization $A/\{A_1, \ldots, A_n\}$ is exactly the set of all atoms that are instances of the atom A but not of any A_i. The model $\mathcal{M}(D)$ represented by a general DIG $D = \{G_1, \ldots, G_m\}$ is the union of the models $\mathcal{M}(\{G_1\}), \ldots, \mathcal{M}(\{G_m\})$.

Example 5.2. Let $\Sigma = (\{P\}, \{s, 0\})$, where 0 is a constant, s is a unary function symbol and P is a binary predicate. Let $D = \{G_1, G_2\}$ be a DIG over Σ, where the two implicit generalizations in D are given by $G_1 = P(s(x), s(y))/\{P(x, x)\}$ and $G_2 = P(0, y)/\{P(x, 0)\}$. The model represented by D is

$$\mathcal{M}(D) = \{P(t, t') \mid t, t' \in \mathcal{T}(\{s, 0\}) \text{ and } t \neq t' \text{ and } t' \neq 0\}\,.$$

$\Diamond$

If an implicit generalization $G = A/\{A_1, \ldots, A_n\}$ contains an atom A_i that cannot be unified with A, then eliminating A_i from G does not change the Herbrand model represented by $\{G\}$. If A_i and A can be unified by a most general unifier σ, then replacing A_i by $A_i\sigma$ in G does not change the Herbrand model represented by $\{G\}$ either.

Without loss of generality, we may thus assume for each implicit generalization $G = A/\{A_1, \ldots, A_n\}$ that all atoms $A_1, \ldots, A_n$ are instances of A. If A is of the form $A = P(\vec{t})$, then we say that G is an implicit generalization *over* P. If $G_1, \ldots, G_n$ are implicit generalizations over $P_1, \ldots, P_n$, respectively, we say that $\{G_1, \ldots, G_n\}$ is a DIG *over* $\{P_1, \ldots, P_n\}$.

5.2 Clausal Representations

We will now translate each DIG D into a set of constrained clauses whose minimal model is $\mathcal{M}(D)$. Our first approach to this translation will result in a set $R_0(D)$ that has the desired minimal model but may also have other Herbrand models. This means that in general $\models_{Ind}$ and $\models_\Sigma$ do not agree for $R_0(D)$. Hence the calculus COR is not complete for $\mathcal{M}(D)$ based on $R_0(D)$ alone. In Section 5.3, we will use the predicate completion procedure from Section 4.2 to enrich $R_0(D)$ by additional constrained clauses, such that the resulting clause set $R(D)$ has exactly one Herbrand model over the given signature.

Definition 5.3. For each DIG D, we define a constrained clause set $R_0(D)$ as follows. If $D = \{G_1, \ldots, G_n\}$ is a DIG over $\{P\}$, let $\dot{P}_1, \check{P}_1, \ldots \dot{P}_n, \check{P}_n$ be fresh predicates. For $G_i = P(\vec{s})/\{P(\vec{s}_1), \ldots, P(\vec{s}_n)\}$, the predicate $\dot{P}_i$ will be used to describe the atom $P(\vec{s})$ and serve as an over-approximation of P, and $\check{P}_i$ will be used to describe the atoms $P(\vec{s}_j)$. Define auxiliary clause sets

$$R_0(G_i) = \{\rightarrow \dot{P}_i(\vec{s}),\ \rightarrow \check{P}_i(\vec{s}_1),\ \ldots,\ \rightarrow \check{P}_i(\vec{s}_n)\}.$$

Then

$$R_0(D) = \bigcup_{1 \le i \le n} R_0(G_i) \cup \{\dot{P}_i(\vec{x}) \to \check{P}_i(\vec{x}), P(\vec{x})\} \ .$$

If $D = D_1 \cup \ldots \cup D_m$ such that each D_i is a DIG over a single predicate and D_i and D_j are DIGs over different predicates whenever $i \neq j$, let

$$R_0(D) = R_0(D_1) \cup \ldots \cup R_0(D_m) \ .$$

We assume that fresh predicates are smaller with respect to $\prec$ than all predicates from the original signature and that $\dot{P}_i \prec \check{P}_j$ for all fresh predicates $\dot{P}_i$ and $\check{P}_j$.

Example 5.4. Consider the DIG D from Example 5.2. The sets $R_0(G_1)$ and $R_0(G_2)$ consist of the following unconstrained clauses:

$$\begin{aligned} R_0(G_1) &= \{\to \dot{P}_1(s(x), s(y)), \quad && \to \check{P}_1(x, x)\} \\ R_0(G_2) &= \{\to \dot{P}_2(0, y), && \to \check{P}_2(x, 0)\} \end{aligned}$$

$R_0(D)$ additionally contains the unconstrained clauses

$$\dot{P}_1(x, y) \to \check{P}_1(x, y), P(x, y) \text{ and}$$
$$\dot{P}_2(x, y) \to \check{P}_2(x, y), P(x, y) \ .$$

Note that each clause in $R_0(D)$ has a unique strictly maximal literal that is positive. Hence $R_0(D)$ is saturated with respect to COR (with a selection function selecting no literals at all) and $\mathcal{I}_{R_0(D)}$ is a minimal Herbrand model of $R_0(D)$ over the extended signature.

Proposition 5.5 (Equivalence of D and $R_0(D)$)**.** *Let D be a DIG and let ϕ be a formula over Σ. Then $\mathcal{M}(D) \models \phi$ iff $R_0(D) \models_{\mathit{Ind}} \phi$.*

Proof. It suffices to show the proposition in the case where $\phi = P(\vec{t})$ is a ground atom. Let $D = \{G_1, \ldots, G_m\}$. Then $\mathcal{M}(D) \models P(\vec{t})$ holds iff there is a G_i such that $\mathcal{M}(\{G_i\}) \models P(\vec{t})$. If we write $G_i = P(\vec{s})/\{P(\vec{s}_1), \ldots, P(\vec{s}_n)\}$, then this is equivalent to $P(\vec{t})$ being an instance of $P(\vec{s})$ but not of any $P(\vec{s}_j)$. This in turn is equivalent to $R_0(\{G_i\}) \models_{\mathit{Ind}} \dot{P}_i(\vec{t})$ and $R_0(\{G_i\}) \not\models_{\mathit{Ind}} \check{P}_i(\vec{t})$. That this holds for some i is equivalent to $R_0(D) \models_{\mathit{Ind}} P(\vec{t})$. $\Diamond$

In general, the set $R_0(D)$ will have more than one Herbrand model over the given signature:

Example 5.6. The set $R_0(D)$ from Example 5.4 has several Herbrand models over the signature $(\{P, \dot{P}_1, \dot{P}_2, \check{P}_1, \check{P}_2\}, \{s, 0\})$. One of them is $\mathcal{M}(D)$, another one is the model in which all of $\dot{P}_1(t, t')$, $\dot{P}_2(t, t')$, $\check{P}_1(t, t')$, $\check{P}_2(t, t')$, and $P(t, t')$ are valid for all ground terms t, t'.

In the next section, we will remedy this by completing the set $R_0(D)$.

5.3 Completed Clausal Representations

Using the predicate completion algorithm from Section 4, it is possible to extend the clause set $R_0(D)$ in order to exclude non-minimal models.

Definition 5.7. Let D be a DIG. The constrained clause set $R(D)$ is defined as the completion of $R_0(D)$, as defined in Section 4.2.

The set $R_0(D)$ is obviously stepwise saturated: All clauses have a strictly maximal positive literal, the new predicates $\dot{P}_i$ and $\check{P}_i$ are of level 1 and the original predicates are of level 2 in $R_0(D)$. Hence Lemma 4.5 implies that $\mathcal{I}_{R_0(D)}$ is in fact the only Herbrand model of $R(D)$:

Lemma 5.8. *Let D be a DIG over $\Sigma = (\mathcal{P}, \mathcal{F})$ and let $\mathcal{P}'$ be the set of fresh predicates in $R_0(D)$. Then $R(D)$ has exactly one Herbrand model over $\Sigma' = (\mathcal{P} \cup \mathcal{P}', \mathcal{F})$, namely $\mathcal{I}_{R_0(D)}$.*

Hence for every formula ϕ over Σ', it holds that $R(D) \models_{Ind} \phi$ iff $R(D) \models_\Sigma \phi$ iff $R(D) \cup \{\neg\phi\}$ is Herbrand-unsatisfiable over Σ'.

Together with Proposition 5.5, this implies that validity in $\mathcal{M}(D)$ is equivalent to validity in all Herbrand models of $R(D)$:

Corollary 5.9 (Equivalence of D and $R(D)$)**.** *Let D be a DIG and let ϕ be a formula over Σ. Then $\mathcal{M}(D) \models \phi$ iff $R(D) \models_\Sigma \phi$.*

Example 5.10. Consider the DIG D and the set $R_0(D)$ from Example 5.4. To compute $R(D)$, we have to look at the sets of clauses defining the predicates $\dot{P}_1, \dot{P}_2, \check{P}_1, \check{P}_2$, and P:

$$\begin{aligned}
N_{\dot{P}_1} &= \{\to \dot{P}_1(s(x), s(y))\} & N_{\check{P}_1} &= \{\to \check{P}_1(x, x)\}\\
N_{\dot{P}_2} &= \{\to \dot{P}_2(0, y)\} & N_{\check{P}_2} &= \{\to \check{P}_2(x, 0)\}\\
N_P &= \{\dot{P}_1(x, y) \to \check{P}_1(x, y), P(x, y)\ , & &\dot{P}_2(x, y) \to \check{P}_2(x, y), P(x, y)\}\ .
\end{aligned}$$

The negation of $\dot{P}_1$ in the minimal model of $R_0(D)$ is obviously defined by $\neg\dot{P}_1(x, y) \iff \neg\exists x', y'. x \simeq s(x') \wedge y \simeq s(y')$. The quantifier elimination procedure simplifies the right hand side $\neg\exists x', y'. x \simeq s(x') \wedge y \simeq s(y')$ of this equivalence to $x \simeq 0 \vee y \simeq 0$. This results in the unconstrained completion

$$N'_{\dot{P}_1} = \{\dot{P}_1(0, y) \to,\ \dot{P}_1(x, 0) \to\}\ .$$

Analogously, the negation of $\check{P}_1$ in the minimal model of $R_0(D)$ is defined by $\neg\check{P}_1(x, y) \iff x \not\simeq y$. The corresponding completion is not unconstrained:

$$N'_{\check{P}_1} = \{x \not\simeq y \parallel \check{P}_1(x, y) \to\}$$

The completions of $\dot{P}_2$ and $\check{P}_2$ are computed analogously as

$$N'_{\dot{P}_2} = \{\dot{P}_2(s(x), y) \rightarrow\} \text{ and}$$
$$N'_{\check{P}_2} = \{\check{P}_2(x, s(y)) \rightarrow\} \ .$$

For P, note that the clauses in N_P could equivalently be written as

$$\dot{P}_1(x, y) \wedge \neg\check{P}_1(x, y) \rightarrow P(x, y) \text{ and}$$
$$\dot{P}_2(x, y) \wedge \neg\check{P}_2(x, y) \rightarrow P(x, y) \ .$$

Hence $\neg P(x, y) \iff (\neg\dot{P}_1(x, y) \vee \check{P}_1(x, y)) \wedge (\neg\dot{P}_2(x, y) \vee \check{P}_2(x, y))$. Rewriting the right hand side of this equivalence to its disjunctive normal form

$$\begin{aligned} &(\check{P}_1(x, y) \wedge \check{P}_2(x, y)) \\ \vee\ &(\check{P}_1(x, y) \wedge \neg\dot{P}_2(x, y)) \\ \vee\ &(\check{P}_2(x, y) \wedge \neg\dot{P}_1(x, y)) \\ \vee\ &(\neg\dot{P}_1(x, y) \wedge \neg\dot{P}_2(x, y)) \end{aligned}$$

forms the basis to translate this definition into the following clause set:

$$\begin{aligned} N'_P = \{&P(x, y), \check{P}_1(x, y), \check{P}_2(x, y) \rightarrow\ , \\ &P(x, y), \check{P}_1(x, y) \rightarrow \dot{P}_2(x, y)\ , \\ &P(x, y), \check{P}_2(x, y) \rightarrow \dot{P}_1(x, y)\ , \\ &P(x, y) \rightarrow \dot{P}_1(x, y), \dot{P}_2(x, y)\} \end{aligned}$$

The set $R(D)$ is then the union of the starting set $R_0(D)$ and all partial completions N'_Q, $Q \in \{\dot{P}_1, \dot{P}_2, \check{P}_1, \check{P}_2, P\}$. ◊

In this example, constraints consisting of disequations appear exactly in the completion of $N_{\check{P}_1} = \{\rightarrow \check{P}_1(x, x)\}$, because the completion has to capture the fact that $P_1(x, y)$ can only be false if x and y are different. In general, such constraints always arise from clauses in which the maximal literal is non-linear, i.e. whenever a variable appears twice in this literal. E.g. the completion of $\{\rightarrow Q(x, x, x)\}$ adds the clauses $x \not\simeq y \parallel Q(x, y, z) \rightarrow$, $x \not\simeq z \parallel Q(x, y, z) \rightarrow$, and $y \not\simeq z \parallel Q(x, y, z) \rightarrow$. Such non-linearities are also the only reason for the appearance of constraint disequations.

Corollary 5.9 implies that we can use the calculus COR to reason about validity in $\mathcal{M}(D)$. In Section 6, we will explore when this approach results in a decision procedure.

6 Decidability Results

Representing DIGs as sets of constrained clauses allows us to deduce various superposition-based decidability results. Because of the simple shape of the clauses in R_0, a lemma by Ganzinger and Stuber [12] guarantees that the inductive validity of ground queries is decidable (Corollary 6.3). More general queries can be decided using COR: For ARMs, the inductive validity of formulas of the form $\forall\vec{x}.\exists\vec{y}.\phi$ and $\exists\vec{x}.\forall\vec{y}.\phi$ with quantifier-free ϕ is decidable (Theorem 6.9). For DIGs, the inductive validity of several subclasses is decidable (Theorem 6.8). This extends results by Fermüller and Pichler, who proved the inductive validity of unconstrained clauses to be decidable [11].

6.1 Decidability of Ground Queries

The question whether a ground query holds in $\mathcal{M}(D)$ is decidable even without completion, using an approach by Ganzinger and Stuber [12]. This approach relies only on the saturation of $R_0(D)$ and its universal reductiveness. A clause $\Gamma \to \Delta$ is *universally reductive* if either $\Delta = \emptyset$ or if $\Delta = \Delta', A$ such that A is strictly maximal in $\Gamma \to \Delta$ and all variables of $\Gamma \to \Delta$ occur in A.

Lemma 6.1 ([12, Lemma 4]). *Let N be a saturated, finite and universally reductive set of clauses. Then it is decidable whether a ground atom A is valid in $\mathcal{I}_N$.*

Proof. If $\mathcal{I}_N \models A$, then there must be a clause $\Gamma \to \Delta, B \in N$ producing A. This is the case iff $A = B\sigma$ is an instance of B (assuming without loss of generality that $B\sigma$ is maximal in $(\Gamma \to \Delta, B)\sigma$), every atom of $\Gamma\sigma$ is true in $\mathcal{I}_N$ and every atom of $\Delta\sigma$ is false in $\mathcal{I}_N$. Because $\Gamma \to \Delta, B$ is universally reductive and $B\sigma$ is ground, every atom of $\Gamma\sigma$ and $\Delta\sigma$ is ground. Moreover, every such atom is strictly smaller than A, so deciding their validity is strictly simpler than deciding the validity of the original query A. $\Diamond$

Because $R_0(D)$ is indeed saturated and universally reductive, it directly follows that ground queries are decidable for DIGs.

Lemma 6.2. *Let D be a DIG. Then all clauses in $R_0(D)$ are universally reductive.*

Proof. All clauses of $R_0(D)$ are either purely positive or of the form

$$\dot{P}_i(\vec{x}) \rightarrow \check{P}_i(\vec{x}), P(\vec{x}) \ ,$$

where the literal $P(\vec{x})$ is strictly maximal. Both types of clauses are obviously universally reductive. ◊

Corollary 6.3. *Let D be a DIG and let A be a ground atom. Then it is decidable whether $\mathcal{M}(D) \models A$.*

6.2 Decidability of DIG Equivalence

Let us investigate in more detail what the constrained clauses in the completion $R(D)$ of $R_0(D)$ look like. Consider first a single implicit generalization $G = P(\vec{t})/\{P(\vec{s}_1), \ldots, P(\vec{s}_n)\}$. All constrained clauses in $R_0(G)$ are unconstrained units. The only clause in $R_0(G)$ defining $\dot{P}$ is $\rightarrow \dot{P}(\vec{t})$, i.e. $\vec{x} \simeq \vec{t} \Longrightarrow \dot{P}(\vec{x})$ is valid in every model of $R_0(G)$. In the minimal model, both implications $\vec{x} \simeq \vec{t} \Longrightarrow \dot{P}(\vec{x})$ and $\vec{x} \simeq \vec{t} \Longleftarrow \dot{P}(\vec{x})$ hold. So the complement of $\dot{P}(\vec{x})$ in the minimal model of $R_0(G)$ is defined by $\neg \dot{P}(\vec{x}) \iff \neg(\vec{x} \simeq \vec{t})$, or (in addition to $\rightarrow \dot{P}(\vec{t})$) by the constrained clauses

$$x_i \not\simeq t_i \parallel \dot{P}(\vec{x}) \rightarrow \ .$$

In the case of $\check{P}$, there are several defining clauses, and the completion consists of the constrained clauses of the form

$$x_{i_1} \not\simeq s_{1,i_1}, \ldots, x_{i_n} \not\simeq s_{n,i_n} \parallel \check{P}(\vec{t}) \rightarrow \ .$$

For a DIG D, $R_0(D)$ contains, in addition to the clauses presented above, only clauses of the form $\dot{P}_i(\vec{x}) \rightarrow \check{P}_i(\vec{x}), P(\vec{x})$, where $P(\vec{x})$ is the maximal literal occurrence. So P is defined in the minimal model of $R_0(D)$ by $P(\vec{x}) \iff \bigvee_{1 \in \{1 \ldots n\}} \dot{P}_i(\vec{x}) \wedge \neg \check{P}_i(\vec{x})$. Its complement is hence defined by $\neg P(\vec{x}) \iff \bigwedge_{1 \in \{1 \ldots n\}} \neg \dot{P}_i(\vec{x}) \vee \check{P}_i(\vec{x})$, or, bringing the right hand side into disjunctive normal form, by

$$\neg P(\vec{x}) \iff \bigvee_{i_1, \ldots, i_n \in \{1 \ldots n\}} \check{P}_{j_1}(\vec{x}) \wedge \ldots \wedge \check{P}_{j_m}(\vec{x}) \wedge \dot{P}_{j_{m+1}}(\vec{x}) \wedge \ldots \wedge \dot{P}_{j_n}(\vec{x}) \ ,$$

where $\dot{P}_{j_1}, \check{P}_{j_1}, \ldots, \dot{P}_{j_n}, \check{P}_{j_n}$ are the fresh predicates introduced for P. The disjuncts correspond to the unconstrained clauses

$$P(\vec{x}), \check{P}_{j_1}(\vec{x}), \ldots, \check{P}_{j_m}(\vec{x}) \rightarrow \dot{P}_{j_{m+1}}(\vec{x}), \ldots, \dot{P}_{j_n}(\vec{x}) \ .$$

Note that all constrained non-unit clauses contain a unique literal that is maximal for all instances of the constrained clause, namely $P(\vec{x})$.

If we restrict the rules of the calculus COR to a certain class of constrained clauses, then all derivable conclusions belong again to this class:

Lemma 6.4. *Let D be a DIG over $\Sigma = (\mathcal{P}, \mathcal{F})$ and let $\mathcal{P}'$ be the set of fresh predicates in $R(D)$. Then the set of constrained clauses over $(\mathcal{P} \cup \mathcal{P}', \mathcal{F})$ of the following forms is closed under the inference rules of COR:*

(1) $\alpha \,\|\, \rightarrow A$ *or* $\alpha \,\|\, A \rightarrow$ *or* $\alpha \,\|\, \Box$

(2) $\alpha \,\|\, \dot{P}_i(\vec{t}) \rightarrow \check{P}_i(\vec{t})$

(3) $\alpha \,\|\, \check{P}_{i_1}(\vec{t}), \ldots, \check{P}_{i_k}(\vec{t}), \dot{P}_{i_{k+1}}(\vec{t}), \ldots, \dot{P}_{i_l}(\vec{t}) \rightarrow \dot{P}_{i_{l+1}}(\vec{t}), \ldots, \dot{P}_{i_m}(\vec{t})$
where each part of the constrained clause may be empty and all predicates have identical term arguments.

Moreover, the saturation of a finite set of such constrained clauses with COR and an empty selection function terminates.

Proof. Ordered factoring inferences can only take constrained clauses of type (3) as premise and obviously yield a constrained clause of type (3) again. Because of the fact that $\dot{P}_i \prec \check{P}_j$ for all $\dot{P}_i, \check{P}_j \in \mathcal{P}'$ and thus ordered resolution primarily works on predicates $\check{P}_j$, one easily checks closure under ordered resolution.

Extend the partial ordering $\prec$ to a complete ordering on $\mathcal{P}'$ and write $\mathcal{P}' = \{Q_1, \ldots, Q_n\}$ such that $Q_{i+1} \prec Q_i$ for all $1 \leq i < n$. Given a constrained clause $\alpha \,\|\, C$, let p_i be the number of positive and q_i the number of negative occurrences of the predicate Q_i in C. In each inference between constrained clauses of the given form, the tuple $(p_1, q_1, \ldots, p_n, q_n)$ is lexicographically strictly smaller for the conclusion than for each premise: This is obvious for factoring inferences. A resolution inference always has the form

$$\frac{\alpha_1 \,\|\, \Gamma_1 \rightarrow \Delta_1, Q_k(\vec{t_1}) \quad \alpha_2 \,\|\, \Gamma_2, Q_k(\vec{t_2}) \rightarrow}{(\alpha_1, \alpha_2^{\not\approx} \,\|\, \Gamma_1, \Gamma_2 \rightarrow \Delta_1, \Delta_2)\sigma_1\sigma_2} \ .$$

Note that all literals in the first (or second, respectively) premise are of the form $Q_i(\vec{t_1})$ (or $Q_i(\vec{t_2})$) with identical argument terms, $Q_k(\vec{t_1})\sigma_1\sigma_2$ is strictly maximal in the first premise and $Q_k(\vec{t}_2)\sigma_1\sigma_2$ is maximal in the second premise. Hence Q_k is the maximal predicate appearing in both premises and it occurs only once in the first premise and only negatively in the second premise. So for the first premise, the first non-zero component of $(p_1, q_1, \ldots, p_n, q_n)$ is $p_k = 1$; for the second premise, it is q_k. For the conclusion, the first possibly non-zero component is q_k, and this component is one smaller for the conclusion than for the second premise.

Hence only finitely many clauses (up to renaming of universal variables) can be derived using COR from a finite set of such constrained clauses.

Note that, while redundancy is undecidable in general, the very restricted notion of redundancy that suffices here, where an inference is redundant if its conclusion or a variant thereof has already been derived, is obviously decidable. $\Diamond$

Lemma 6.5. *Let D be a DIG over $\Sigma = (\mathcal{P}, \mathcal{F})$ and let $\mathcal{P}'$ be the set of fresh predicates in $R(D)$. If N is a set of constrained clauses over Σ containing at most one literal each, then it is decidable whether $R(D) \cup N$ is Herbrand-satisfiable over $\Sigma' = (\mathcal{P} \cup \mathcal{P}', \mathcal{F})$.*

Proof. Let M^* be a saturation of $M = R(D) \cup N$ by the calculus COR with a selection function that does not select any literals. By Theorem 3.9, Herbrand-unsatisfiability of M over Σ' is equivalent to the coverage of A_{M^*}, which is decidable if M^* is finite.

To prove that M^* is finite, we show that any derivation starting from M is finite. The only constrained clauses containing at least two literals and a predicate symbol of $\mathcal{P}$ are of the form $\dot{P}_i(\vec{x}) \rightarrow P(\vec{x}), \check{P}_i(\vec{x})$ or of the form $P(\vec{x}), \check{P}_{j_1}(\vec{x}), \ldots, \check{P}_{j_m}(\vec{x}) \rightarrow \dot{P}_{j_{m+1}}(\vec{x}), \ldots, \dot{P}_{j_n}(\vec{x})$, where $P \in \mathcal{P}$, $\dot{P}_{j_1}, \ldots, \dot{P}_{j_n}, \check{P}_{j_1}, \ldots, \check{P}_{j_n} \in \mathcal{P}'$ are the fresh predicates introduced for P. Note that for each i, either $\dot{P}_{j_i}(\vec{x})$ or $\check{P}_{j_i}(\vec{x})$ occurs in each constrained clause of the latter type, and $P(\vec{x})$ is the maximal literal occurrence in both types of constrained clauses (cf. the initial remarks in this Section). Since each inference between constrained clauses containing predicate symbols of $\mathcal{P}$ reduces the number of atoms featuring such a predicate, there are only finitely many such inferences.

The conclusion of an inference

$$\frac{\| \, \dot{P}_i(\vec{x}) \rightarrow P(\vec{x}), \check{P}_i(\vec{x}) \qquad \| \, P(\vec{x}), \check{P}_{j_1}(\vec{x}), \ldots, \check{P}_{j_m}(\vec{x}) \rightarrow \dot{P}_{j_{m+1}}(\vec{x}), \ldots, \dot{P}_{j_n}(\vec{x})}{\| \, \Gamma \rightarrow \Delta}$$

between two constrained clauses in $R(D)$ using $P \in \mathcal{P}$ is a tautology (and thus redundant), because either $\dot{P}_i(\vec{x})$ or $\check{P}_i(\vec{x})$ appears in both Γ and Δ. The remaining derivable constrained clauses over $(\mathcal{P}', \Sigma)$ obey the restrictions of Lemma 6.4, hence the saturation terminates. $\Diamond$

With this preliminary work done, we can decide whether two DIGs represent the same model:

Theorem 6.6 (DIG Equivalence)**.** *Equivalence of DIGs is decidable by ordered resolution.*

Proof. Let D, D' be two DIGs. Because $\mathcal{M}(D) = \bigcup_{G \in D} \mathcal{M}(\{G\})$, and because $\mathcal{M}(D) = \mathcal{M}(D')$ iff $\mathcal{M}(D) \subseteq \mathcal{M}(D')$ and $\mathcal{M}(D') \subseteq \mathcal{M}(D)$, it suffices to show the decidability of $\mathcal{M}(D) \subseteq \mathcal{M}(D')$ in the case where $D = \{G\}$ consists of a single implicit generalization $G = P(\vec{s})/\{P(\vec{s}\sigma_1), \ldots, P(\vec{s}\sigma_n)\}$. Without loss of generality, we assume that $P(\vec{s})$ and $P(\vec{s}\sigma_1), \ldots, P(\vec{s}\sigma_n)$ do not share any variables.

Let $x_1, \ldots, x_m$ be the variables in $P(\vec{s})$ and let $y_1, \ldots, y_k$ be the variables in $P(\vec{s}\sigma_1), \ldots, P(\vec{s}\sigma_n)$. The implicit generalization G states that the formula $\forall \vec{x}.(\forall \vec{y}.\vec{x} \not\simeq \vec{x}\sigma_1 \wedge \ldots \wedge \vec{x} \not\simeq \vec{x}\sigma_n) \implies P(\vec{s})$ holds in $\mathcal{M}(D)$.

By Proposition 5.9, $\mathcal{M}(D) \subseteq \mathcal{M}(D')$ holds iff $R(D') \models_\Sigma P(\vec{t})$ for every atom $P(\vec{t}) \in \mathcal{M}(D)$. Equivalently, the set $R(D') \cup \{\exists \vec{x}.\forall \vec{y}.\vec{x} \not\simeq \vec{x}\sigma_1 \wedge \ldots \wedge \vec{x} \not\simeq \vec{x}\sigma_n \wedge \neg P(\vec{s})\}$ does not have a Herbrand model over Σ. Writing the latter formula as a set of constrained clauses, we can derive that the same holds for the constrained clause set $R(D') \cup \{\vec{v} \simeq \vec{x}\sigma_1 \parallel \Box, \ \ldots, \ \vec{v} \simeq \vec{x}\sigma_n \parallel \Box, \ \vec{v} \simeq \vec{x} \parallel P(s) \rightarrow\}$.

By Lemma 6.5, whether this constrained clause set has a Herbrand model over Σ is decidable by means of the calculus COR. ◊

Example 6.7. The DIG $D' = \{P(x, s(y))/\{P(s(x'), s(x'))\}\}$ and the DIG D from Examples 5.2 and 5.10 describe the same model. We only show that $\mathcal{M}(D) \supseteq \mathcal{M}(D')$.

Expressed as a satisfiability problem of constrained clauses, we have to check whether $R(D) \cup \{v_1 \simeq x, v_2 \simeq y \parallel P(x, s(y)) \rightarrow, \ v_1 \simeq s(x'), v_2 \simeq x' \parallel \Box\}$ is Herbrand-satisfiable over Σ. To do so, we saturate this set with respect to COR.

Since $R(D) \cup \{v_1 \simeq s(x'), v_2 \simeq x' \parallel \Box\}$ is saturated, all non-redundant inferences use at least one descendant of $v_1 \simeq x, v_2 \simeq y \parallel P(x, s(y)) \rightarrow$. The following constrained clauses can be derived. We index the new constrained clauses by (0)... (9). Each of these constrained clauses is derived from one clause in $R(D)$ (which is not repeated here) and another clause that is indicated by its index:

index	constrained clause	derived from
(0)	$v_1 \simeq s(x'), v_2 \simeq x' \parallel \quad \square$	
(1)	$v_1 \simeq x, v_2 \simeq y \parallel P(x, s(y)) \to$	
(2)	$v_1 \simeq x, v_2 \simeq y \parallel \dot{P}_1(x, s(y)) \to \check{P}_1(x, s(y))$	(1)
(3)	$v_1 \simeq x, v_2 \simeq y \parallel \dot{P}_2(x, s(y)) \to \check{P}_2(x, s(y))$	(1)
(4)	$v_1 \simeq s(x), v_2 \simeq y \parallel \quad \to \check{P}_1(s(x), s(y))$	(2)
(5)	$v_1 \simeq s(x), v_2 \simeq y, x \not\simeq s(y) \parallel \dot{P}_1(x, s(y)) \to$	(2)
(6)	$v_1 \simeq s(x), v_2 \simeq y, s(x) \not\simeq s(y) \parallel \quad \square$	(4) or (5)
(7)	$v_1 \simeq 0, v_2 \simeq y \parallel \quad \to \check{P}_2(0, s(y))$	(3)
(8)	$v_1 \simeq x, v_2 \simeq y \parallel \dot{P}_2(x, s(y)) \to$	(3)
(9)	$v_1 \simeq 0, v_2 \simeq y \parallel \quad \square$	(7) or (8)

No further non-redundant constrained clauses can be derived. The constraint set

$$\{(v_1 \simeq s(x'), v_2 \simeq x'), (v_1 \simeq s(x), v_2 \simeq y, s(x) \not\simeq s(y)), (v_1 \simeq 0, v_2 \simeq y)\}$$

consisting of the constraints of the constrained clauses (0), (6), and (9) is covering, which means that the whole constrained clause set is Herbrand-unsatisfiable over Σ, i.e. that $\mathcal{M}(D) \supseteq \mathcal{M}(D')$. $\Diamond$

6.3 Decidability of Formula Entailment

Apart from deciding equivalence of DIG representations, we can decide for formulas from a number of classes whether they are true in models represented by DIGs.

Theorem 6.8 (Decidability of DIG Formula Entailment)**.** *Let D be a DIG and let ϕ be a quantifier-free formula over Σ with variables $\vec{x}, \vec{y}$. The following problems are decidable:*

(1) $\mathcal{M}(D) \models \forall \vec{x}. \exists \vec{y}. \phi$ is decidable if one of the following holds:

- *(a) ϕ is a single clause*
- *(b) ϕ is a conjunction of clauses of the form $\to \Delta$*
- *(c) ϕ is a conjunction of clauses of the form $\Gamma \to$*
- *(d) ϕ is a conjunction of unit clauses where no predicate appears in both a positive and a negative literal*

(2) $\mathcal{M}(D) \models \exists \vec{x}. \forall \vec{y}. \phi$ is decidable if one of the following holds:

(a) *ϕ is a conjunction of literals*

(b) *ϕ is a conjunction of clauses of the form $\Gamma \rightarrow$*

(c) *ϕ is a conjunction of clauses of the form $\rightarrow \Delta$*

(d) *ϕ is a clause where no predicate appears in both a positive and a negative literal*

Proof. We first consider the case (1a).
Let $\Sigma = (\mathcal{P}, \mathcal{F})$ and let $\mathcal{P}'$ be the set of fresh predicates in $R(D)$. Let $\phi = C = A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ and let $N = \{\vec{v} \simeq \vec{x} \parallel \rightarrow A_1,\ \ldots,\ \vec{v} \simeq \vec{x} \parallel \rightarrow A_n,\ \vec{v} \simeq \vec{x} \parallel B_1 \rightarrow,\ \ldots,\ \vec{v} \simeq \vec{x} \parallel B_n \rightarrow\}$. By Proposition 5.9, $\mathcal{M}(D) \models \forall \vec{x}. \exists \vec{y}. \phi$ is equivalent to $R(D) \models_\Sigma \forall \vec{x}. \exists \vec{y}. \phi$. This in turn is equivalent to the Herbrand-unsatisfiability of $R(D) \cup \{\exists \vec{x}. \forall \vec{y}. \neg \phi\}$, or equivalently $R(D) \cup N$, over the signature $\Sigma' = (\mathcal{P} \cup \mathcal{P}', \mathcal{F})$. By Lemma 6.5, the Herbrand-unsatisfiability of $R(D) \cup N$ over Σ' is decidable.

The proofs for (1b)–(1d) are exactly analogous, using slight variations of Lemma 6.5. The decidability of the problems (2a)–(2d) reduces to (1a)–(1d), respectively, because $\mathcal{M}(D) \models \exists \vec{x}. \forall \vec{y}. \phi$ if and only if $\mathcal{M}(D) \not\models \forall \vec{x}. \exists \vec{y}. \neg \phi$. ◊

The simple nature of atomic representations allows us to go one step further:

Theorem 6.9 (Decidability of ARM Formula Entailment)**.** *Let D be an ARM over Σ and let ϕ be a quantifier-free formula over Σ with variables $\vec{x}, \vec{y}$. It is decidable whether $\mathcal{M}(D) \models \forall \vec{x}. \exists \vec{y}. \phi$ and whether $\mathcal{M}(D) \models \exists \vec{x}. \forall \vec{y}. \phi$.*

Proof. We first show the decidability of $\mathcal{M}(D) \models \forall \vec{x}. \exists \vec{y}. \phi$. Let $\Sigma = (\mathcal{P}, \mathcal{F})$ and let $\mathcal{P}'$ be the set of fresh predicates in $R(D)$. We write the formula $\neg \phi$ as an equivalent finite set $N_{\neg \phi}$ of unconstrained clauses and set $N = \{(\vec{v} \simeq \vec{x} \parallel C) \mid C \in N_{\neg \phi}\}$.

Consider first some unconstrained clause $C = A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ and assume that constrained clauses in $R(D)$ and C do not share any universal variables. It holds that $\mathcal{M}(D) \not\models C$ iff there are constrained clauses $\alpha_i \parallel \rightarrow A'_i$ and $\beta_j \parallel B'_j \rightarrow$ in $R(D)$ and a substitution $\tau : X \rightarrow \mathcal{T}(\mathcal{F})$ such that $A_i \tau = A'_i \tau$, $B_j \tau = B'_j \tau$ and $\alpha_i \tau$ and $\beta_j \tau$ are satisfiable[1] for all i, j. By definition of $R(D)$, all positive clauses in $R(D)$ are unconstrained, so this is equivalent to the formula $\bigvee \beta \tau$ being satisfiable, where the disjunction ranges over all $\beta = \beta_1, \ldots, \beta_k$ and τ such that there are constrained clauses $\parallel \rightarrow A'_i$ and $\beta_j \parallel B'_j \rightarrow$ in $R(D)$ and τ is a most general simultaneous unifier of all (A_i, A'_i) and (B_j, B'_j).

[1]Note that all β_j are purely negative and so none of them contains any existential variables.

Coming back to the validity of N, it holds that $\mathcal{M}(D) \not\models N$ iff for every substitution $\sigma : V \to \mathcal{T}(\mathcal{F})$ there is a substitution $\tau : \text{vars}(\alpha \,\|\, C) \setminus V \to \mathcal{T}(\mathcal{F})$ and a constrained clause $\alpha \,\|\, C \in N$, such that $\alpha\sigma\tau$ is satisfiable and $\mathcal{M} \not\models C\tau$. By the considerations above, this is equivalent to the satisfiability of the formula $\bigwedge_{\alpha \,\|\, C \in N} \bigvee \alpha\tau \wedge \beta\tau$.

$\mathcal{M}(D) \models \exists\vec{x}.\forall\vec{y}.\phi$ is decided analogously, without negating ϕ. $\Diamond$

7 Conclusion

We have extended the decidability results of [11] for ARMs to arbitrary formulas with one quantifier alternation and for DIGs to several more restrictive formula structures with one quantifier alternation.

Our approach has potential for further research. We restricted our attention to a non-equational setting, whereas our initial fixed domain calculus [14] considers equations as well. It is an open problem to what extend our results also hold in an equational setting. In [11], the finite and infinite (open) signature semantics for DIGs was considered. Our results refer to the finite signature semantics where actually only the signature symbols of a finite saturated set are considered in the minimal model. It is not known what an infinite (further symbols) signature semantics means to our approach. Finally, in [11] the question was raised what happens if one considers more restrictive, e.g., linear DIGs. We know that linear DIGs require less effort in predicate completion but it is an open question whether this has further effects on decidability (complexity) results.

Bibliography

[1] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994. Revised version of Technical Report MPI-I-91-208, 1991.

[2] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier, 2001.

[3] P. Baumgartner and C. Tinelli. The model evolution calculus. In *Proceedings of the 19th International Conference on Automated Deduction (CADE)*, pages 350–364, 2003.

[4] R. Caferra, A. Leitsch, and N. Peltier. *Automated Model Building*, volume 31 of *Applied Logic Series*. Kluwer, 2004.

[5] K. L. Clark. Negation as failure. In *Logic and Data Bases*, pages 293–322, 1977.

[6] H. Comon. Disunification: A survey. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 322–359. MIT Press, Cambridge, MA, 1991.

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

[8] H. Comon and R. Nieuwenhuis. Induction = I-axiomatization + first-order consistency. *Information and Computation*, 159(1/2):151–186, May 2000.

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

[10] C. G. Fermüller, A. Leitsch, U. Hustadt, and T. Tamet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 25, pages 1791–1849. Elsevier, 2001.

[11] C. G. Fermüller and R. Pichler. Model representation over finite and infinite signatures. *Journal of Logic and Computation*, 17(3):453–477, 2007.

[12] H. Ganzinger and J. Stuber. Inductive theorem proving by consistency for first-order clauses. In M. Rusinowitch and J. Rémy, editors, *CTRS 92*, volume 656 of *LNCS*, pages 226–241, 1992.

[13] T. Hillenbrand and C. Weidenbach. Superposition for finite domains. Research Report MPI-I-2007-RG1-002, Max-Planck Institute for Informatics, Saarbruecken, Germany, April 2007.

[14] M. Horbach and C. Weidenbach. Superposition for fixed domains. In M. Kaminski and S. Martini, editors, *Proc. of the 17th Annual Conference of the European Association for Computer Science Logic, CSL 2008*, volume 5213 of *Lecture Notes in Computer Science*, pages 293–307, Berlin / Heidelberg, September 2008. Springer.

[15] M. Horbach and C. Weidenbach. Decidability results for saturation-based model building. In R. Schmidt, editor, *Proc. of the 22nd International Conference on Automated Deduction, CADE-22*, volume 5663 of *LNAI*, pages 404–420, Berlin / Heidelberg, August 2009. Springer.

[16] U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.

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

[18] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier, 2001.

[19] C. Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 27, pages 1965–2012. Elsevier, 2001.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available via WWW using the URL `http://www.mpi-inf.mpg.de`. If you have any questions concerning WWW access, please contact `reports@mpi-inf.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-inf.mpg.de`

MPI-I-2009-RG1-002	P. Wischnewski, C. Weidenbach	Contextual rewriting
MPI-I-2009-5-006	S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, G. Weikum	Scalable phrase mining for ad-hoc text analytics
MPI-I-2009-5-004	N. Preda, F.M. Suchanek, G. Kasneci, T. Neumann, G. Weikum	Coupling knowledge bases and web services for active knowledge
MPI-I-2009-5-003	T. Neumann, G. Weikum	The RDF-3X engine for scalable management of RDF data
MPI-I-2008-RG1-001	A. Fietzke, C. Weidenbach	Labelled splitting
MPI-I-2008-5-004	F. Suchanek, M. Sozio, G. Weikum	SOFI: a self-organizing framework for information extraction
MPI-I-2008-5-003	F.M. Suchanek, G. de Melo, A. Pease	Integrating Yago into the suggested upper merged ontology
MPI-I-2008-5-002	T. Neumann, G. Moerkotte	Single phase construction of optimal DAG-structured QEPs
MPI-I-2008-5-001	F. Suchanek, G. Kasneci, M. Ramanath, M. Sozio, G. Weikum	STAR: Steiner tree approximation in relationship-graphs
MPI-I-2008-4-003	T. Schultz, H. Theisel, H. Seidel	Crease surfaces: from theory to extraction and application to diffusion tensor MRI
MPI-I-2008-4-002	W. Saleem, D. Wang, A. Belyaev, H. Seidel	Estimating complexity of 3D shapes using view similarity
MPI-I-2008-1-001	D. Ajwani, I. Malinger, U. Meyer, S. Toledo	Characterizing the performance of Flash memory storage devices and its impact on algorithm design
MPI-I-2007-RG1-002	T. Hillenbrand, C. Weidenbach	Superposition for finite domains
MPI-I-2007-5-003	F.M. Suchanek, G. Kasneci, G. Weikum	Yago : a large ontology from Wikipedia and WordNet
MPI-I-2007-5-002	K. Berberich, S. Bedathur, T. Neumann, G. Weikum	A time machine for text search
MPI-I-2007-5-001	G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum	NAGA: searching and ranking knowledge
MPI-I-2007-4-008	J. Gall, T. Brox, B. Rosenhahn, H. Seidel	Global stochastic optimization for robust and accurate human motion capture
MPI-I-2007-4-007	R. Herzog, V. Havran, K. Myszkowski, H. Seidel	Global illumination using photon ray splatting
MPI-I-2007-4-006	C. Dyken, G. Ziegler, C. Theobalt, H. Seidel	GPU marching cubes on shader model 3.0 and 4.0
MPI-I-2007-4-005	T. Schultz, J. Weickert, H. Seidel	A higher-order structure tensor
MPI-I-2007-4-004	C. Stoll, E. de Aguiar, C. Theobalt, H. Seidel	A volumetric approach to interactive shape editing
MPI-I-2007-4-003	R. Bargmann, V. Blanz, H. Seidel	A nonlinear viseme model for triphone-based speech synthesis
MPI-I-2007-4-002	T. Langer, H. Seidel	Construction of smooth maps with mean value coordinates
MPI-I-2007-4-001	J. Gall, B. Rosenhahn, H. Seidel	Clustered stochastic optimization for object recognition and pose estimation

MPI-I-2007-2-001	A. Podelski, S. Wagner	A method and a tool for automatic veriication of region stability for hybrid systems
MPI-I-2007-1-003	A. Gidenstam, M. Papatriantafilou	LFthreads: a lock-free thread library
MPI-I-2007-1-002	E. Althaus, S. Canzar	A Lagrangian relaxation approach for the multiple sequence alignment problem
MPI-I-2007-1-001	E. Berberich, L. Kettner	Linear-time reordering in a sweep-line algorithm for algebraic curves intersecting in a common point
MPI-I-2006-5-006	G. Kasnec, F.M. Suchanek, G. Weikum	Yago - a core of semantic knowledge
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A neighborhood-based approach for clustering of linked document collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining linguistic and statistical analysis to extract relations from web documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment texture editing in monocular video sequences based on color-coded printing patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: index-access optimized top-k query processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-aware global df estimation in distributed information retrieval systems
MPI-I-2006-4-010	A. Belyaev, T. Langer, H. Seidel	Mean value coordinates for arbitrary spherical polygons and polyhedra in $\mathbb{R}^3$
MPI-I-2006-4-009	J. Gall, J. Potthoff, B. Rosenhahn, C. Schnoerr, H. Seidel	Interacting and annealing particle filters: mathematics and a recipe for applications
MPI-I-2006-4-008	I. Albrecht, M. Kipp, M. Neff, H. Seidel	Gesture modeling and animation by imitation
MPI-I-2006-4-007	O. Schall, A. Belyaev, H. Seidel	Feature-preserving non-local denoising of static and time-varying range data
MPI-I-2006-4-006	C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, H. Seidel	Enhanced dynamic reflectometry for relightable free-viewpoint video
MPI-I-2006-4-005	A. Belyaev, H. Seidel, S. Yoshizawa	Skeleton-driven laplacian mesh deformations
MPI-I-2006-4-004	V. Havran, R. Herzog, H. Seidel	On fast construction of spatial hierarchies for ray tracing
MPI-I-2006-4-003	E. de Aguiar, R. Zayer, C. Theobalt, M. Magnor, H. Seidel	A framework for natural animation of digitized models
MPI-I-2006-4-002	G. Ziegler, A. Tevs, C. Theobalt, H. Seidel	GPU point list generation through histogram pyramids
MPI-I-2006-4-001	A. Efremov, R. Mantiuk, K. Myszkowski, H. Seidel	Design and evaluation of backward compatible high dynamic range video compression
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On verifying complex properties using symbolic shape analysis
MPI-I-2006-1-007	H. Bast, I. Weber, C.W. Mortensen	Output-sensitive autocompletion search
MPI-I-2006-1-006	M. Kerber	Division-free computation of subresultants using bezout matrices
MPI-I-2006-1-005	A. Eigenwillig, L. Kettner, N. Wolpert	Snap rounding of Bézier curves
MPI-I-2006-1-004	S. Funke, S. Laue, R. Naujoks, L. Zvi	Power assignment problems in wireless communication
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated retraining methods for document classification and their parameter tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An emperical model for heterogeneous translucent objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric calibration of high dynamic range cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A. Magnor, H. Seidel	Joint motion and reflectance capture for creating relightable 3D videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and design of discrete normals and curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse meshing of uncertain and noisy surface scattered data