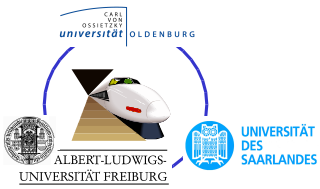

AVACS – Automatic Verification and Analysis of Complex Systems

REPORTS

of SFB/TR 14 AVACS

Editors: Board of SFB/TR 14 AVACS

Obtaining Finite Local Theory Axiomatizations via Saturation

by
Matthias Horbach and Viorica Sofronie-Stokkermans

AVACS Technical Report No. 93
March 2014
ISSN: 1860-9821

Publisher: Sonderforschungsbereich/Transregio 14 AVACS
(Automatic Verification and Analysis of Complex Systems)
Editors: Bernd Becker, Werner Damm, Bernd Finkbeiner, Martin Fränzle,
Ernst-Rüdiger Olderog, Andreas Podelski
ATRs (AVACS Technical Reports) are freely downloadable from `www.avacs.org`

Author(s) contact: Viorica Sofronie-Stokkermans (`vss@avacs.org`).

Obtaining Finite Local Theory Axiomatizations via Saturation

Matthias Horbach and Viorica Sofronie-Stokkermans

University Koblenz-Landau and Max-Planck-Institut für Informatik Saarbrücken

Abstract. In this paper we present a method for obtaining local sets of clauses from possibly non-local ones. For this, we follow the work of Basin and Ganzinger and use saturation under a version of ordered resolution. In order to address the fact that saturation can generate infinite sets of clauses, we use constrained clauses and show that a link can be established between saturation and locality also for constrained clauses: This often allows us to give a finite representation of possibly infinite saturated sets of clauses.

1 Introduction

Many problems in mathematics and computer science can be reduced to proving the satisfiability of conjunctions of literals in a background theory. It is therefore very important to identify situations where reasoning in extensions and combinations of theories can be done efficiently and accurately. The most important issues which need to be addressed in this context are:

(i) finding possibilities of reducing the search space without losing completeness,
(ii) making modular or hierarchical reasoning possible.

In [24], we introduced a class of theory extensions (which we named local) for which both aspects above can be addressed: (i) complete instantiation schemes exist, and (ii) hierarchical and modular reasoning is possible. However, locality is a property of an axiomatization of a theory rather than of the theory itself. Therefore, it is very important to *recognize locality* of a set of clauses, and *to obtain local axiomatizations* – for instance by transforming non-local sets of clauses into local ones. In [4, 5], Basin and Ganzinger presented a link between (order)-locality and saturation under ordered (hyper)resolution. Their result allows to obtain, by saturation, local axiomatizations for a theory from non-local ones. The drawback is that the size of the saturated sets of clauses is often very large and sometimes the saturation process may not terminate. The main contributions of this paper are:

- In order to obtain finite representations of possibly infinite sets of clauses we use *constrained clauses*.
- We use a sound and complete ordered resolution and superposition calculus for constrained clauses. In cases when a classical saturation process might not terminate, the use of constrained clauses allows us to give a finite representation for possibly infinite sets of clauses.

- We show that for certain types of constrained clauses a link can be established between saturation in our calculus and order locality.
- We indicate the limitations of our approach.

In [18] we identified situations in which the combination of two local theory extensions of a base theory $\mathcal{T}_0$ is a local extension of $\mathcal{T}_0$. The assumptions on the component theories are syntactic, so can be easily checked. Together with the results presented in this paper, this allows to prove the locality of combinations of theories, or to obtain local axiomatizations for combinations of theories.

We briefly discuss the relationships between our results and existing work.

Ordered resolution and superposition are often used to devise decision procedures for various theories, cf. e.g. [2, 19, 1]. Our approach allows us to consider, in addition, situations in which the saturated sets are not finite, by giving finite representations for them.

The connection between saturation under ordered resolution and (order) locality was studied by Basin and Ganzinger in [4, 5]. It is however relatively easy to see that no link between saturation under superposition and (order) locality can be established in general. Therefore, if we start with a set N of clauses in first-order logic with equality, then in order to use the results in [4, 5] for proving the locality of N (or for constructing an equivalent local axiomatization) we usually need to saturate $N \cup EQ$ under ordered resolution (where EQ is the set of congruence axioms). The results presented in this paper allow us to identify a class of clauses in first-order logic with equality for which we can prove locality (or construct equivalent local axiomatizations) using superposition – without having to explicitly take into account the congruence axioms.

Constrained clauses are often used in automated theorem proving in order to restrict the number of instances to be considered or for defining a notion of schematic saturation (cf. e.g. [22, 20, 21, 26]). In contrast with this type of results, the constraints we use here are formulae, equalities between variables (to which substitutions are applied) or more general so-called regular constraints. These constraints generalize regular expressions and allow infinite sets of clauses with a repeating structure to be captured by a single constrained clause. The use of the s^+ operation for reasoning about integer offsets in [26] is similar. The difference is that whereas in our work the regular expressions occur in constraints, in [26] they occur in the main clauses, and the constraints only ensure that some of the variables can only be instantiated with constants.

Structure of the paper. The paper is structured as follows: In Sect. 2 we introduce the terminology used in the paper and present the main results on local theory extensions. In Sect. 3 we give ways of recognizing locality; then explain the problem we address in this paper and the idea of our solution. In Sect. 4 we define a constraint inference calculus which we use in Sect. 5 for giving finite saturations of infinite sets of clauses (where we also discuss the limitations of this approach).

This paper is an extended version of an article appearing at FroCoS 2013 [13].

2 Preliminaries

In this section we introduce the terminology and main results used in the paper.

2.1 General Definitions

We build on the notions of [3, 27, 16] and shortly recall here the most important concepts concerning terms and orderings and the specific extensions (concerning constrained clauses) needed in this article. To keep the presentation concise, we restrict ourselves to single-sorted signatures. The many-sorted case works similarly, and it is explicitly taken into account in the later sections.

Terms and Clauses. Let $\Pi = (\Sigma, \mathsf{Pred})$ be a *signature* consisting of a set Σ of function symbols of fixed arity and a set Pred of predicate symbols of fixed arity, and let X be an infinite set of variables such that X and Σ are disjoint.

We denote by $\mathcal{T}_\Sigma(X)$ the set of all *terms* over Σ and X and by $\mathcal{T}_\Sigma$ the set of all *ground terms* over Σ. To improve readability, term tuples $(t_1, \ldots, t_n)$ will often be denoted by $\vec{t}$. The variables occurring in a term t or a term tuple $\vec{t}$ are denoted by $\mathrm{vars}(t)$ or $\mathrm{vars}(\vec{t})$, respectively.

An equation is a multiset of two terms $t_1, t_2 \in \mathcal{T}_\Sigma(X)$, usually written $t_1 \approx t_2$. A *predicative atom* is an expression of the form $P(t_1, \ldots, t_n)$, where $P \in \mathsf{Pred}$ is a predicate symbol of arity n and $t_1, \ldots, t_n \in \mathcal{T}_\Sigma(X)$ are terms. An *atom* is an equation or a predicative atom. 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. If $C = A_1, \ldots, A_n \to B_1, \ldots, B_m$ is a ground clause, we denote by $\neg C$ the set of unit Horn clauses $\to A_i$ and $B_j \to$.

Orderings. A (strict partial) *ordering* $\prec$ on a set S is a transitive and irreflexive binary relation on S. It is *total* if $s \prec t$ or $t \prec s$ whenever $s \neq t$. It is *well-founded* if there is no infinite descending chain $s_1 \succ s_2 \succ \ldots$ of elements of S.

An ordering $\prec$ on $\mathcal{T}_\Sigma(X)$ has the *subterm property* if $t \succ t'$ whenever t contains t' as a strict subterm. It is *stable under substitutions* if $t \prec t'$ implies $t\sigma \prec t'\sigma$ for all t, t' and all substitutions σ. It is a *reduction ordering* if it is well-founded, has the subterm property, and is stable under substitutions.

Let $\prec_\mathcal{T}$ be an ordering on $\mathcal{T}_\Sigma(X)$ and let $\prec$ be an ordering on atoms over $\mathcal{T}_\Sigma(X)$. Then $\prec$ is *compatible* with $\prec_\mathcal{T}$ if for all atoms A_1, A_2 it holds that $A_1 \prec A_2$ if every term in A_1 is bounded by a term in A_2, i.e. if for each term t_1 in A_1 there is a term t_2 in A_2 such that $t_1 \prec_\mathcal{T} t_2$. Note that for finite signatures of predicate symbols, any total ordering on atoms that is compatible with a total and well-founded term ordering is well-founded.

Any ordering $\prec$ on atoms can be extended to 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 in a clause C is *maximal* 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* if there is no other occurrence of an atom in C that is greater than or equal to the occurrence of A w.r.t. $\prec$.

Let $\prec_\mathcal{T}$ be a term ordering and $\prec$ be an atom ordering. A ground clause is *reductive (w.r.t. $\prec_\mathcal{T}$ and $\prec$)* if all of its $\prec_\mathcal{T}$-maximal terms appear in the $\prec$-maximal atoms.[1]

Inferences. A (clausal) *inference rule* is a relation on clauses. Its elements are called *inferences* and written as

$$\frac{C_1 \dots \; C_k}{C} \quad .$$

The clauses $C_1, \dots, C_k$ are called the *premises* and C the *conclusion* of the inference. An *inference calculus* is a set of inference rules. In what follows we will use the standard inference rules for ordered resolution and hyperresolution (as well extensions to constrained clauses in Section 4.2).

Ordered resolution with selection. Let S be a selection function which selects in any clause a subset of negative literals. The ordered resolution calculus with selection consists of the following inference rules:

Ordered Resolution with Selection:

$$\frac{\Gamma_1 \to \Delta_1, A_1, \quad \Gamma_2, A_2 \to \Delta_2}{(\Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2)\sigma}$$

where (1) σ is the most general unifier of A_1 and A_2, (2) $A_1\sigma$ is strictly maximal in $(\Gamma_1 \to \Delta_1, A_1)\sigma$ and nothing is selected in $\Gamma_1 \to \Delta_1, A_1$, (3) A_2 is selected in $\Gamma_2, A_2 \to \Delta_2$ or else $A_2\sigma$ is maximal in $(\Gamma_2, A_2 \to \Delta_2)\sigma$ and nothing is selected in $\Gamma_2, A_2 \to \Delta_2$.

Ordered Factoring:

$$\frac{\Gamma \to \Delta, A_1, A_2}{(\Gamma \to \Delta, A)\sigma}$$

where (1) σ is the most general unifier of A_1 and A_2, (2) $A_1\sigma$ is maximal in $(\Gamma \to \Delta, A_1, A_2)\sigma$ and nothing is selected in $\Gamma \to \Delta, A_1$.

Ordered hyperresolution with selection. Ordered hyperresolution with selection consists of the ordered hyperresolution rule with selection and ordered factoring with selection:

[1] It can be seen that a ground clause C is reductive w.r.t. $\prec_\mathcal{T}$ and $\prec$ if and only if (if A is the maximal atom in C under $\prec$ then for each atom B in C and each term t in B there exists a term s in A such that $s \preceq_\mathcal{T} t$ in the term ordering). The latter is the definition used in [5].

Ordered Hyperresolution with Selection:

$$\frac{\Gamma_1 \to \Delta_1, A_1, \quad \ldots, \quad \Gamma_n \to \Delta_n, A_n \qquad B_1, \ldots, B_n, \Gamma \to \Delta}{(\Gamma_1, \ldots, \Gamma_n, \Gamma \to \Delta_1, \ldots \Delta_n, \Delta)\sigma}$$

where (1) σ is the most general unifier of the unification problem $\{A_1 = B_1, \ldots, A_n = B_n\}$, (2) $A_i\sigma$ is strictly maximal in $(\Gamma_i \to \Delta_i, A_i)\sigma$ and nothing is selected in $\Gamma_i \to \Delta_i, A_i$ for $i = 1, \ldots, n$, (3) $n \geq 1$ and the selected atoms in the last premise are $A_1, \ldots, A_n$ or else $n = 1$, nothing is selected in the last premise and $B_1\sigma$ is maximal in $(B_1, \Gamma \to \Delta)\sigma$.

For Horn clauses the rule above will have the form:

$$\frac{\Gamma_1 \to A_1, \quad \ldots, \quad \Gamma_n \to A_n \qquad B_1, \ldots, B_n, \Gamma \to \Delta}{(\Gamma_1, \ldots, \Gamma_n, \Gamma \to \Delta)\sigma}$$

where (1) σ is the most general unifier of the unification problem $\{A_1 = B_1, \ldots, A_n = B_n\}$, (2) $A_i\sigma$ is strictly maximal in $(\Gamma_i \to \Delta_i, A_i)\sigma$ and nothing is selected in $\Gamma_i \to A_i$ for $i = 1, \ldots, n$, (3) $n \geq 1$ and the selected atoms in the last premise are $A_1, \ldots, A_n$ or else $n = 1$, nothing is selected in the last premise and $B_1\sigma$ is maximal in $(B_1, \Gamma \to \Delta)\sigma$.

Redundancy. Let N be a set of clauses. A ground clause C (not necessarily a clause in N) is redundant in N (w.r.t. $\succ$) if it is entailed by the instances of N which are smaller than C. An inference is called *redundant* w.r.t. N if its conclusion is redundant w.r.t. N or if a premise C is redundant w.r.t. $N \setminus \{C\}$. A clause set N is *saturated* (w.r.t. a given inference calculus) if each inference with premises in N is redundant w.r.t. N. A *derivation* is a finite or infinite sequence $N_0, N_1, \ldots$ such that for each i, there is an inference with premises in N_i and conclusion C that is not redundant w.r.t. N_i, such that $N_{i+1} = N_i \cup \{C\}$.

2.2 Local Theories

The notion of local set of Horn clauses (or local Horn theory) was introduced by Givan and McAllester in [9, 11]. A *local set of Horn clauses* is a set of Horn clauses $\mathcal{K}$ such that, for any ground Horn clause C, $\mathcal{K} \models C$ only if already $\mathcal{K}[C] \models C$ (where $\mathcal{K}[C]$ is the set of instances of $\mathcal{K}$ in which all terms are subterms of ground terms in either $\mathcal{K}$ or $\mathcal{C}$). The size of $\mathcal{K}[G]$ is polynomial in the size of G for a fixed $\mathcal{K}$. Since satisfiability of sets of ground Horn clauses can be checked in linear time [6], it follows that for local Horn theories, validity of ground Horn clauses can be checked in polynomial time. Givan and McAllester proved that every problem which is decidable in PTIME can be encoded as an entailment problem of ground clauses w.r.t. a local Horn theory [10]. An example of a local Horn theory (cf. [10]) is the set of axioms of a monotone function w.r.t. a transitive relation $\leq$, consisting of the following set of (implicitly universally quantified) Horn clauses:

$$\mathcal{K} = \{x \leq y \ \wedge \ y \leq z \ \to \ x \leq z, \quad x \leq y \to f(x) \leq f(y)\}.$$

Another example provided in [10] is a local axiom set for reasoning about a lattice (similar to that proposed by Skolem in [23]).

Order locality. In [4, 5], Basin and Ganzinger defined the more general notion of *order locality*. Given a term ordering $\prec$, we say that a set $\mathcal{K}$ of clauses entails a ground clause C bounded by $\prec$ (notation: $\mathcal{K} \models_{\preceq} C$), if and only if there is a proof of $\mathcal{K} \models C$ from those ground instances of clauses in $\mathcal{K}$ in which (under $\preceq$) each term is smaller than or equal to some term in C. A set of clauses $\mathcal{K}$ is *local with respect to* $\prec$ if whenever $\mathcal{K} \models C$ for a ground clause C, then $\mathcal{K} \models_{\preceq} C$. They also showed how to recognize (order-)local theories and how to use these results for automated complexity analysis. Further details on this are presented in Section 3.

2.3 Local Theory Extensions

In [8, 24] the notion of locality for Horn clauses is extended to the notion of *local extension* of a base theory.

Let $\Pi_0{=}(\Sigma_0, \mathsf{Pred})$ be a signature, and $\mathcal{T}_0$ be a theory with signature Π_0. We here consider extensions $\mathcal{T} := \mathcal{T}_0 \cup \mathcal{K}$ of $\mathcal{T}_0$ with new function symbols Σ (called *extension functions*) whose properties are axiomatized using a set $\mathcal{K}$ of (universally closed) clauses in the extended signature $\Pi = (\Sigma_0 \cup \Sigma, \mathsf{Pred})$.

Example 1 *Let $\mathcal{T}_0$ be a theory of integers with signature containing the unary function s and the predicate symbol $\leq$. Let $\Sigma = \{f\}$ where f is a new function symbol. Consider the following sets of clauses:*

- $\mathcal{K}_f^1 = \{\forall x, y\, (x \leq y \rightarrow f(x) \leq f(y))\}$,
- $\mathcal{K}_f^2 = \{\forall x (f(x) \leq f(s(x)))\}$

(both axiomatizations for the monotonicity of f). For $i = 1, 2$, $\mathcal{T}_i := \mathcal{T}_0 \cup \mathcal{K}_f^i$ is an extension of $\mathcal{T}_0$ with function f satisfying the set $\mathcal{K}_f^i$ of clauses.

Our goal is to address proof tasks of the form $G \models_{\mathcal{T}_0 \cup \mathcal{K}} \bot$ (written also: $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$) where G is a set of ground clauses with additional (fresh) constants (in a countable set C), i.e. in the signature $\Pi^C = (\Pi_0 \cup \Sigma)^C = (\Sigma_0 \cup \Sigma \cup C, \mathsf{Pred})$.

Locality conditions. Let $\mathcal{T}_0$ be an arbitrary theory with signature $\Pi_0 = (\Sigma_0, \mathsf{Pred})$, where the set of function symbols is Σ_0. Let $\Pi = (\Sigma_0 \cup \Sigma, \mathsf{Pred}) \supseteq \Pi_0$ be an extension by a non-empty set Σ of new function symbols and $\mathcal{K}$ be a set of (implicitly universally closed) clauses in the extended signature. Let C be a fixed countable set of fresh constants.

Notation: Let T be a set of ground terms in the signature Π^C. We denote by $\mathcal{K}[T]$ the set of all instances of $\mathcal{K}$ in which the terms starting with a function symbol in Σ are in T. Formally:

$$\mathcal{K}[T] := \{\varphi\sigma \mid \forall \bar{x}.\, \varphi(\bar{x}) \in \mathcal{K},\ \text{where (i) if } f \in \Sigma \text{ and } t = f(t_1, ..., t_n) \text{ occurs in } \varphi\sigma \text{ then } t \in T\text{; (ii) if } x \text{ is a variable that does not appear below some } \Sigma\text{-function in } \varphi \text{ then } \sigma(x) = x\}.$$

An extension $\mathcal{T}_0 \cup \mathcal{K}$ of $\mathcal{T}_0$ is *local* if it satisfies the following condition[2]:

(Loc) For every set G of ground clauses in Π^C it holds that $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$ if and only if $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G \models \bot$

where $\mathcal{K}[G] = \mathcal{K}[\mathsf{est}(\mathcal{K}, G)]$ consists of those instances of $\mathcal{K}$ in which the terms starting with *extension functions* are in the set $\mathsf{est}(\mathcal{K}, G)$ of extension ground terms (i.e. terms starting with a function in Σ) which already occur in G or $\mathcal{K}$. In [17] we generalized condition (Loc) by considering closure operators on ground terms. Let Ψ be a closure operator associating with every set T of ground terms a set $\Psi(T)$ of ground terms. For any set G of ground Π^C-clauses we write $\mathcal{K}[\Psi_{\mathcal{K}}(G)]$ for $\mathcal{K}[\Psi(\mathsf{est}(\mathcal{K}, G))]$. We define versions of locality in which the set of terms used in the instances of the axioms is described using the map Ψ:

(Loc^{Ψ}) For every set G of ground clauses in Π^C it holds that $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$ if and only if $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G \models \bot$.

Extensions satisfying condition (Loc^{Ψ}) are called Ψ-local. A *finite* locality condition (Loc^{Ψ}_f) is defined by restricting the locality conditions to hold for *finite* sets G of ground clauses.

Local theory extensions are Ψ-local, where Ψ=id (the identity operator). Local theories can be seen as local extensions of the "empty theory" (the pure theory of equality, with empty signature). The *order-local theories* introduced in [5] satisfy a $\Psi^{\preceq}$-locality condition: Here, if T is a set of ground clauses, $\Psi^{\preceq}(T)$ is defined as

$$\Psi^{\preceq}(T) = \{s \mid s \text{ is a ground term and } s \preceq t \text{ for some } t \in T\}$$

for a given ordering $\prec$ on ground terms [5]. In this case we write also $\mathcal{K}[\preceq T]$ for $\mathcal{K}[\Psi^{\preceq}(T)]$. If $T = \mathsf{st}(\mathcal{K}, G)$ is the set of all ground terms occurring in $\mathcal{K}$ or in G, we also use the notation $\mathcal{K}[\preceq G]$. Thus (with the notation introduced in Section 2.2) if C is a ground clause then $\mathcal{K} \models_{\preceq} C$ iff $\mathcal{K}[\preceq C] \models C$. If $\prec$ is the subterm relationship then $\mathcal{K}[\preceq G] = \mathcal{K}[G]$.

Hierarchical reasoning. Let $\mathcal{T}_0 \subseteq \mathcal{T} = \mathcal{T}_0 \cup \mathcal{K}$ be a theory extension satisfying condition (Loc^{Ψ}). To check the satisfiability w.r.t. $\mathcal{T}$ of a set G of ground Π^C-clauses, we proceed as follows: By locality, $\mathcal{T} \cup G \models \bot$ iff $\mathcal{T}_0 \cup \mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G \models \bot$. We purify $\mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ by introducing, in a bottom-up manner, new constants c_t for subterms $t = f(g_1, \ldots, g_n)$ with $f \in \Sigma$, g_i ground $(\Sigma_0 \cup C)$-terms, together with their definitions $c_t \approx t$. The set of formulae thus obtained has the form $\mathcal{K}_0 \cup G_0 \cup D$, where D consists of definitions of the form $c \approx f(g_1, \ldots, g_n)$, where $f \in \Sigma$, c is a constant, $g_1, \ldots, g_n$ are ground $(\Sigma_0 \cup C)$-terms, and $\mathcal{K}_0, G_0$ are Π^C_0-formulae. We reduce the problem to testing satisfiability in $\mathcal{T}_0$ as follows:

[2] It is easy to check that the formulation we give here and that in [24] are equivalent.

Theorem 2 ([24]) *Let $\mathcal{K}$ and G be as specified above. Assume that $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ satisfies condition (Loc^{Ψ}). Let $\mathcal{K}_0 \cup G_0 \cup D$ be obtained from $\mathcal{K}[\Psi_{\mathcal{K}}(G)] \cup G$ as explained above. Then*

$$\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot \text{ if and only if } \mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0 \models \bot$$

where

$$\mathsf{Con}_0 = \left\{ \bigwedge_{i=1}^{n} c_i \approx d_i \rightarrow c = d \;\middle|\; \begin{array}{l} c \approx f(c_1, \ldots, c_n) \in D, \\ d \approx f(d_1, \ldots, d_n) \in D \end{array} \right\} .$$

If $\mathcal{K}[\Psi_{\mathcal{K}}(G)]$ is finite and $\mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0$ belongs to a decidable fragment of $\mathcal{T}_0$ then we can effectively check the satisfiability of G w.r.t. $\mathcal{T}_0 \cup \mathcal{K}$.

3 Recognizing Ψ-local Theory Extensions

We present several ways of recognizing local and Ψ-local theory extensions.

3.1 Locality and Embedability

Links between *locality of a theory* and *embeddability* of partial models into total ones were established in [7]. Similar results can also be obtained for *local theory extensions*. When establishing links between locality and embeddability, we require that the extension clauses in $\mathcal{K}$ are *flat* and *linear* w.r.t. Σ-functions:

A *(non-ground) extension clause* D is *Σ-flat* when all symbols below a Σ-function symbol in D are variables. D is *Σ-linear* if, whenever a variable occurs in two terms of D which start with Σ-functions, the terms are identical, and no term starting with a Σ-function contains two occurrences of a variable.

Let $\Pi = (\Sigma, \mathsf{Pred})$ be a first-order signature with set of function symbols Σ and set of predicate symbols Pred. A *partial Π-structure* is a structure $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \Sigma}, \{P_{\mathcal{A}}\}_{P \in \mathsf{Pred}})$, where A is a non-empty set, for every $f \in \Sigma$ with arity n, $f_{\mathcal{A}}$ is a partial function from A^n to A, and for every $P \in \mathsf{Pred}$, $P_{\mathcal{A}} \subseteq A^n$. We consider constants (0-ary functions) to be always defined. $\mathcal{A}$ is called a *total structure* if the functions $f_{\mathcal{A}}$ are all total. Given a (total or partial) Π-structure $\mathcal{A}$ and $\Pi_0 \subseteq \Pi$ we denote the reduct of $\mathcal{A}$ to Π_0 by $\mathcal{A}|_{\Pi_0}$. (For the precise definition of a weak partial model for a set of clauses see e.g. [24, 18].) If $\mathcal{T} = \mathcal{T}_0 \cup \mathcal{K}$ is an extension of a Π_0-theory $\mathcal{T}_0$ with new function symbols in Σ and clauses $\mathcal{K}$, we denote by $\mathsf{PMod}_w^{\Psi}(\Sigma, \mathcal{T})$ the set of weak partial models $\mathcal{A}$ of $\mathcal{T}$ whose Σ_0-functions are total, and all terms in $\Psi_{\mathcal{K}}(\mathcal{D}(\mathcal{A})) := \Psi(\mathsf{est}(\mathcal{K}) \cup \{f(a_1, \ldots, a_n) \mid f \in \Sigma, f_{\mathcal{A}}(a_1, \ldots, a_n) \text{ is defined}\}$ are defined – in the extended structure $\mathcal{A}^A$ with constants from $\mathcal{A}$. In [24, 17, 18] we considered embeddability properties of partial algebras, e.g. (Emb_w^{Ψ}) and ($\mathsf{Emb}_{w,f}^{\Psi}$).

(Emb_w^{Ψ}) Every $\mathcal{A} \in \mathsf{PMod}_w^{\Psi}(\Sigma, \mathcal{T})$ weakly embeds into a total model of $\mathcal{T}$.

Condition ($\mathsf{Emb}^{\Psi}_{w,f}$) requires embeddability only for partial algebras where the extension functions have a *finite* domain of definition. We proved that if $\mathcal{K}$ is a set of Σ-flat and Σ-linear clauses in the signature Π and all weak partial models of an extension $\mathcal{T}_0 \cup \mathcal{K}$ of a base theory $\mathcal{T}_0$ with total Σ_0-functions can be embedded into a total model of the extension, then the extension is local (i.e. that (Emb^{Ψ}_{w}) implies (Loc^{Ψ})) [24, 17, 18]. Conversely, we showed that if $\mathcal{K}$ is a set of Σ-flat clauses in the signature Π then if $\mathcal{T}_0$ is a first-order theory and the extension $\mathcal{T}_0 \subseteq \mathcal{T}{=}\mathcal{T}_0{\cup}\mathcal{K}$ satisfies (Loc^{Ψ}) then every model in $\mathsf{PMod}^{\Psi}_{w}(\Sigma, \mathcal{T})$ weakly embeds into a total model of $\mathcal{T}$.

Example 3 *Let $\mathcal{T}_0$ be the theory of integers with successor (s) and ordering $\leq$ described by the model $(\mathbb{N}, s, \leq)$. Let f be a new function symbol.*

- $\mathcal{K}^1_f = \{x \leq y \rightarrow f(x) \leq f(y)\}$ *satisfies condition* $\mathsf{Emb}^{\mathsf{id}}{}_{w}$ *hence defines a local theory extension.*
- *Neither* $\mathcal{K}^2_f = \{f(x){\leq}f(s(x))\}$ *nor its flattened version* $\{y{\approx}s(x){\rightarrow}f(x){\leq}f(y)\}$ *satisfy* $\mathsf{Emb}^{\mathsf{id}}_{\mathsf{w}}$ *(there are weak partial models of this axiom which cannot be extended to total ones – for instance the partial model $\mathcal{A}$ with support $\mathbb{N}$ for which $f_{\mathcal{A}}(2) = 4, f_{\mathcal{A}}(4) = 2$ and f is undefined everywhere else).*

In [18] we introduce a stronger notion of weak embeddability:

(EEmb_w) For every $\mathcal{A} \in \mathsf{PMod}_w(\Sigma, \mathcal{T})$ there is a total model $\mathcal{B}$ of $\mathcal{T}$ and a weak embedding $\varphi : \mathcal{A} \rightarrow \mathcal{B}$ such that the embedding $\varphi : \mathcal{A}|_{\Pi_0} \rightarrow \mathcal{B}|_{\Pi_0}$ is elementary.

The definition generalizes in a natural way to a notion (EEmb^{Ψ}_{w}), parameterized by a closure term operator Ψ by requiring that the embeddability condition holds for all $\mathcal{A} \in \mathsf{PMod}^{\Psi}_{w}(\Sigma, \mathcal{T})$ with domain of definition closed under Ψ, and to corresponding finite embeddability conditions ($\mathsf{EEmb}^{\Psi}_{w,f}$) analogous to ($\mathsf{Emb}^{\Psi}_{w,f}$).

3.2 Locality Transfer

In [18] we analyzed situations in which locality of certain theory extensions can be proved as a consequence of the locality of other extensions.

Locality transfer for combinations of local theory extensions. We consider combinations of Ψ_i-local extensions (with different Ψ_i's) over a common base theory.

Theorem 4 ([18]) *Let $\mathcal{T}_0$ be a theory in the signature Π_0, and let Σ_1 and Σ_2 two disjoint sets of fresh function symbols. For $i = 1, 2$ let $\Pi_i =:= (\Sigma_0 \cup \Sigma_i, \mathsf{Pred})$; let $\mathcal{K}_i$ be a set of universally closed Π_i-clauses and $\mathcal{T}_i := \mathcal{T}_0 \cup \mathcal{K}_i$; and let Ψ_i be term closure operators on ground Π^C_i-terms. Assume that*

(1) $\mathcal{T}_0$ is a $\forall\exists$ theory,
(2) $\mathcal{K}_i$ is Σ_i-flat and $\mathcal{T}_0 \subseteq \mathcal{T}_i$ satisfies condition ($\mathsf{Emb}^{\Psi_i}_{w}$) for $i = 1, 2$,

(3) all variables are shielded in $\mathcal{K}_i$, i.e., all variables occur below an extension function, $i = 1, 2$.

Then $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}_1 \cup \mathcal{K}_2$ has $(\mathsf{Emb}_w^{\Psi_1 \cup \Psi_2})$ where $(\Psi_1 \cup \Psi_2)(\Gamma) := \Psi_1(\Gamma) \cup \Psi_2(\Gamma)$. In particular, if $\mathcal{K}_i$ are (quasi)-flat and linear then extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}_1 \cup \mathcal{K}_2$ is $\Psi_1 \cup \Psi_2$-local.

The restrictions (1)–(3) can be relaxed if one of the component theories say $\mathcal{T}_i$ enjoys a version of the weak embeddability property (Emb_w) in which it is guaranteed that every weak partial model P of $\mathcal{T}_0 \cup \mathcal{K}_i$ with totally defined Π_0 operations weakly embeds into a total model A of $\mathcal{T}_0 \cup \mathcal{K}_i$ with the additional property that the reducts to Π_0 of P and A are elementary equivalent (e.g. satisfy the same Π_0-formulae). The restrictions (1)–(3) are not needed at all for the conclusion of Theorem 4 to hold if both theories have the property above (for details see [18]).

Other locality transfer results. We now present some examples of local theory extensions $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ with the property that locality of the extension is preserved if we "enrich" the structure of $\mathcal{T}_0$.

Theorem 5 ([18]) *Let $\Pi_0 = (\Sigma_0, \mathsf{Pred})$ be a signature, $\mathcal{T}_0$ a theory in Π_0, Σ_1 and Σ_2 two disjoint sets of new function symbols, $\Pi_i := (\Sigma_0 \cup \Sigma_i, \mathsf{Pred})$, $i = 1, 2$. Assume that:*

(C1) *$\mathcal{T}_2$ is a Π_2-theory with $\mathcal{T}_0 \subseteq \mathcal{T}_2$.*
(C2) *$\mathcal{K}$ is a set of universally closed Π_1-clauses.*
(C3) *The extension $\mathcal{T}_0 \subseteq \mathcal{T}_0 \cup \mathcal{K}$ satisfies the embeddability condition ($\mathsf{EEmb}_{w,f}$).*

Then the extension $\mathcal{T}_2 \subseteq \mathcal{T}_2 \cup \mathcal{K}$ also satisfies condition ($\mathsf{EEmb}_{w,f}$). In particular, if $\mathcal{K}$ is (quasi)-flat and linear then extension $\mathcal{T}_2 \subseteq \mathcal{T}_2 \cup \mathcal{K}$ is local.

If all variables in clauses in $\mathcal{K}$ occur below Σ_1-functions, and ground satisfiability is decidable in $\mathcal{T}_2$, then ground satisfiability is decidable in $\mathcal{T}_2 \cup \mathcal{K}$. Theorem 5 is a very useful result, which allows us to identify a large number of local extensions. In [18] we illustrated its applicability on the following example:

Example 6 *Let Lat be the theory of lattices and $\mathcal{T}_1 = \mathsf{Lat} \cup \mathsf{Mon}_f$, where $\mathsf{Mon}_f = \{\forall x, y\ (x \leq y \rightarrow f(x) \leq f(y))\}$ is the monotonicity of a new function symbol f. We can prove that the extension $\mathsf{Lat} \subseteq \mathsf{Lat} \cup \mathsf{Mon}_f$ satisfies condition (C3) in the statement of Theorem 5. Let $\mathcal{T}$ be any extension of the theory of lattices (this can be the theory of distributive lattices, Heyting algebras, Boolean algebras, any theory with a total order – e.g. the (ordered) theory of integers or of reals, etc.). By Theorem 5, $\mathcal{T} \subseteq \mathcal{T} \cup \mathsf{Mon}_f$ satisfies condition ($\mathsf{EEmb}_{w,f}$), hence is local.*

3.3 Locality and Saturation

In [4, 5] conditions for automatically checking *order locality* of sets of clauses are given. As mentioned in Section 2.2, given a term ordering $\prec$, we say that a

set $\mathcal{K}$ of clauses entails a clause C bounded by $\prec$ (notation: $\mathcal{K} \models_{\preceq} C$), if and only if there is a proof of $\mathcal{K} \models C$ from those ground instances of clauses in $\mathcal{K}$ in which (under $\prec$) each term is smaller than or equal to some term in C. As mentioned in Section 2.3, $\mathcal{K} \models_{\preceq} C$ iff $\mathcal{K}[\preceq C] \models C$. Thus (cf. [4, 5]) a set of clauses $\mathcal{K}$ is *local with respect to* $\prec$ if for every ground clause C, $\mathcal{K} \models C$ if and only if $\mathcal{K}[\preceq C] \models C$.

It is not difficult to check that for every set $\mathcal{K}$ of clauses and for every term ordering $\prec$ the following are equivalent:

(1) For every ground clause C: $\mathcal{K} \models C$ if and only if $\mathcal{K}[\preceq C] \models C$.
(2) For every (finite) set of ground clause G: ($\mathcal{K} \cup G$ is unsatisfiable) if and only if ($\mathcal{K}[\preceq G] \cup G$ is unsatisfiable).

In [4, 5], Ganzinger and Basin established a link between peak saturation and order locality, and used these results for automated complexity analysis. They then also considered an ordered hyperresolution calculus for Horn clauses with a selection function which selects in every clause C the set of all negative atoms of C which contain a term which is maximal in C w.r.t. $\prec$ [5]. In [4, 5] the following terminology is used in this case: *Negative* (resp. *positive*) premises are premises in which the literal resolved upon is negative (resp. positive). *Peak inferences* are inferences with the property that for every term t in the conclusion there is a larger term $t' \succ t$ in the negative premise. *Plateau inferences* are inferences for which in the succedent of a negative premise there exists an occurrence of a maximal term. A set of Horn clauses $\mathcal{K}$ is *peak saturated* if all peak inferences are redundant for which (i) the second, negative premise is in $\mathcal{K}$, and (ii) the first, positive premise's antecedent does not contain a maximal term, and (iii) the positive premise is in $\mathcal{K}$ or generated from $\mathcal{K}$ using plateau inferences.

Theorem 7 ([4, 5]) *Let $\prec_{\mathcal{T}}$ be a well-founded (possibly partial) term ordering and $\prec$ a compatible and total atom ordering in first-order logic without equality. Let $\mathcal{K}$ be a set of clauses which is reductive w.r.t. $\prec_{\mathcal{T}}$ and $\prec$.*

- *If $\mathcal{K}$ is saturated w.r.t. $\prec$-ordered resolution, then $\mathcal{K}$ is order local w.r.t. $\prec$.*
- *Let $\mathcal{K}$ be a set of Horn clauses. $\mathcal{K}$ is peak saturated w.r.t. $\prec$-ordered hyperresolution with selection (cf. [5]) if and only if $\mathcal{K}$ is order local w.r.t. $\prec$.*

Theorem 7 allows us to obtain, by saturation, local axiomatizations from non-local ones. We can saturate $\mathcal{K}$ under peak redundancy by first adding all clauses obtained by plateau inferences between clauses in this set (obtaining a set P), and then the conclusions of all peak inferences with negative premise in $\mathcal{K}$ and positive premises of the corresponding form which are in $\mathcal{K}$ or P. Theorem 4 and Corollary 5 identify situations in which we know that combinations of local presentations are again local, or situations in which we can enrich the base theory without loss of locality.

When using saturation to detect locality (or to generate local presentations from non-local ones) one drawback is that equality cannot be used as a built-in predicate: If the clauses contain the equality predicate then the congruence

axioms have to be added explicitly, which can be inefficient. Another drawback is that the size of the saturated sets of clauses can be very large. Often, in fact, infinitely many clauses are generated.

Example 8 *Consider* $\mathsf{Pre} \cup \{f(x) \leq f(s(x))\}$*, where*

$$\mathsf{Pre} = \{x \leq x, \quad x \leq y \wedge y \leq z \rightarrow x \leq z\} \ .$$

By saturation we obtain the infinite set[3]*:* $\{f(x) \leq f(s^n(x)) \mid n \geq 0\} \cup \mathsf{Pre}$*. In such cases, a usual resolution-based theorem prover will not be able to detect saturation, because the set of clauses which are generated is infinite.*

Our goal is to obtain finite representations of possibly infinite sets of clauses. For this, we will use constrained clauses. As a by-product, the form of the constraints may allow us to (conservatively) extend the language – e.g. by defining new predicates – in order to obtain local presentations.

4 A Constrained Inference Calculus

To reduce the size of a representation, we employ constrained clauses. A constrained clause $\alpha \parallel C$ denotes all ground clauses $C\sigma$ where σ satisfies the constraint α. To reason about such constrained clauses, we use the standard inference rules for constrained ordered resolution and superposition (cf. [16]), enriched by an on-the-fly induction rule called *melting*.

4.1 Definitions

Substitution Expressions. A *(basic) substitution* σ is a map from a finite set $X' \subseteq X$ of variables to $\mathcal{T}_\Sigma(X)$. The application of σ to a term t or a term tuple $\vec{t}$ is denoted by $t\sigma$ or $\vec{t}\sigma$, respectively. The substitution σ is *linear* if no variable occurs twice in the term set $\{x\sigma \mid x \in X'\}$.

Substitution expressions are built over substitutions and constructors $\circ$ (composition), $|$ (disjunction), and * (loop) of arity 2, 2 and 1, respectively. Substitution expressions are denoted as $\bar{\sigma}, \bar{\tau}$. The symbols $\circ$ and $|$ are written in infix notation, and * is written in postfix notation. We will often write $\bar{\sigma} \circ \bar{\tau}$ as $\bar{\sigma}\bar{\tau}$.

The *domain* $\mathrm{dom}(\bar{\sigma})$ and the *variable range* $\mathrm{VRan}(\bar{\sigma})$ of a substitution expression are defined as follows: For a substitution $\sigma : \{x_1, \ldots, x_n\} \rightarrow \mathcal{T}_\Sigma(X)$, we define $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$ and $\mathrm{VRan}(\sigma) = \mathrm{vars}(x_1\sigma, \ldots, x_n\sigma)$. For complex expressions, we have

$$\begin{array}{rlrl} \mathrm{dom}(\bar{\sigma} \circ \bar{\tau}) &= \mathrm{dom}(\bar{\sigma}) & \mathrm{VRan}(\bar{\sigma} \circ \bar{\tau}) &= \mathrm{VRan}(\bar{\tau}) \\ \mathrm{dom}(\bar{\sigma}_1 | \bar{\sigma}_2) &= \mathrm{dom}(\bar{\sigma}_1) \cup \mathrm{dom}(\bar{\sigma}_2) & \mathrm{VRan}(\bar{\sigma}_1 | \bar{\sigma}_2) &= \mathrm{VRan}(\bar{\sigma}_1) \cap \mathrm{VRan}(\bar{\sigma}_2) \\ \mathrm{dom}(\bar{\sigma}^*) &= \mathrm{dom}(\bar{\sigma}) & \mathrm{VRan}(\bar{\sigma}^*) &= \mathrm{dom}(\bar{\sigma}) \cup \mathrm{VRan}(\bar{\sigma}) \end{array}$$

A substitution expression $\bar{\sigma}$ is *well-formed*, if (i) for each subexpression $\bar{\tau}_1 \circ \bar{\tau}_2$ of $\bar{\sigma}$, it holds that $\mathrm{VRan}(\bar{\tau}_1) = \mathrm{dom}(\bar{\tau}_2)$, (ii) for each subexpression $\bar{\tau}_1 | \bar{\tau}_2$ of $\bar{\sigma}$, it holds that $\mathrm{dom}(\bar{\tau}_1) = \mathrm{dom}(\bar{\tau}_2)$ and $\mathrm{VRan}(\bar{\tau}_1) = \mathrm{VRan}(\bar{\tau}_2)$, and (iii) for each subexpression $\bar{\tau}^*$ of $\bar{\sigma}$, it holds that $\mathrm{VRan}(\bar{\tau}) = \mathrm{dom}(\bar{\tau})$.

[3] Similarly if we consider the flattened version of $\mathcal{K}_f^2$: $\{y = s(x) \rightarrow f(x) \leq f(y)\}$.

Constrained Clauses. A *constrained clause* $\alpha \parallel C$ consists of a clause C and a *regular constraint* α of the form $(x_1 \approx y_1, \ldots, x_n \approx y_n)\bar{\sigma}$, also written as $(\vec{x} \approx \vec{y})\bar{\sigma}$, such that x_i, y_i are variables and $\bar{\sigma}$ is a well-formed substitution expression with domain $\{x_1, y_1, \ldots, x_n, y_n\}$. If a regular constraint α does not contain any equations, we call $\alpha \parallel C$ *unconstrained* and identify it with its clausal part C.

If $\alpha = (\vec{x} \approx \vec{y})\bar{\tau}$ is a regular constraint, then $\alpha\sigma$ is defined as $(\vec{x} \approx \vec{y})\bar{\tau}\sigma'$, where $\sigma' : \mathrm{VRan}(\bar{\tau}) \to \mathcal{T}(\Sigma, X)$ maps z to $z\sigma$ if $z \in \mathrm{dom}(\sigma)$ and to z otherwise. The application $(\alpha \parallel C)\sigma$ of a substitution to a constrained clause is then defined as $\alpha\sigma \parallel C\sigma$.

The set of *ground instances* of a constrained clause $\alpha \parallel C$ consists of all ground clauses D for which there is a substitution σ such that $C\sigma = D$ and $\alpha\sigma$ is a satisfiable ground constraint. This means that regular constraints are interpreted syntactically.

Orderings. Let $\prec_{\mathcal{T}}$ be a term ordering and $\prec$ be an atom ordering. An (unconstrained) ground clause is *reductive (w.r.t. $\prec_{\mathcal{T}}$ and $\prec$)* if all of its $\prec_{\mathcal{T}}$-maximal terms appear in the $\prec$-maximal atoms. A constrained clause is *reductive (w.r.t. $\prec_{\mathcal{T}}$ and $\prec$)* if all its ground instances are reductive (cf. [5]).

Denotations and Models. We define the *denotation* $[\![\bar{\sigma}]\!]$ of a substitution expression $\bar{\sigma}$ inductively as follows:

$$\begin{aligned} [\![\sigma]\!] &= \{\sigma\} \\ [\![\bar{\sigma}\bar{\tau}]\!] &= \{\sigma\tau \mid \sigma \in [\![\bar{\sigma}]\!], \tau \in [\![\bar{\tau}]\!]\} \\ [\![\bar{\sigma}_1|\bar{\sigma}_2]\!] &= [\![\bar{\sigma}_1]\!] \cup [\![\bar{\sigma}_2]\!] \\ [\![\bar{\sigma}^*]\!] &= \bigcup_{n\geq 0} [\![\bar{\sigma}^n]\!] \end{aligned}$$

Here $\bar{\sigma}^0$ denotes the substitution $\{x \mapsto x \mid x \in \mathrm{dom}\,\bar{\sigma}\}$, and $\bar{\sigma}^{n+1} = \bar{\sigma} \circ \bar{\sigma}^n$.

The semantics of the application of substitution expressions to terms and clauses and the semantics of constrained clause sets are defined just as one would expect by identifying a substitution expression with its denotation and by identifying a constrained clause $(\vec{x} \approx \vec{y})\bar{\sigma} \parallel C$ with the (potentially infinite) clause set $\{\vec{x}\sigma \approx \vec{y}\sigma \to C \mid \sigma \in [\![\bar{\sigma}]\!]\}$ (cf. [16] for details). An interpretation $\mathcal{I}$ is said to *model* a constrained clause set N, written $\mathcal{I} \models N$, if and only if $\mathcal{I} \models C$ for each C in the denotation of a constrained clause in N. In this case, $\mathcal{I}$ is called a *model* of N. A constrained clause set is *satisfiable* if it has a model.

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

$$\frac{\alpha_1 \parallel C_1 \quad \ldots \quad \alpha_k \parallel C_k}{\alpha \parallel C} \quad .$$

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

A constrained clause $(\vec{x}\approx\vec{y})\bar{\sigma} \parallel C$ is *redundant* w.r.t. a constrained clause set N if all of its ground instances are redundant in the sense of Section 2.1 or if there is a variant $(\vec{x}\approx\vec{y})\bar{\tau} \parallel C$ of a constrained clause in N such that $[\![\bar{\sigma}]\!] \subseteq [\![\bar{\tau}]\!]$. An inference is called *redundant* w.r.t. N if its conclusion is redundant w.r.t. N or if a premise C is redundant w.r.t. $N \setminus \{C\}$. A constrained clause set N is *saturated* (w.r.t. a given inference calculus) if each inference with premises in N is redundant w.r.t. N.

4.2 The Melting Calculus for Constrained Clauses

As mentioned before, we use the standard inference rules for constrained ordered resolution and superposition to reason about constrained clauses. The ordered resolution rule for constrained clauses, for example, is:

Ordered Resolution with Selection:

$$\frac{\alpha_1 \parallel \Gamma_1 \to \Delta_1, A_1 \quad \alpha_2 \parallel \Gamma_2, A_2 \to \Delta_2}{(\alpha_1, \alpha_2 \parallel \Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2)\sigma}$$

where
(1) σ is the most general unifier of A_1 and A_2 and nothing is selected in $\Gamma_1 \to \Delta_1, A_1$, (2) $A_1\sigma$ is strictly maximal in $(\Gamma_1 \to \Delta_1, A_1)\sigma$ and (3) A_2 is selected in $\Gamma_2, A_2 \to \Delta_2$ or else $A_2\sigma$ is maximal in $(\Gamma_2, A_2 \to \Delta_2)\sigma$ and nothing is selected in $\Gamma_2, A_2 \to \Delta_2$.

In addition, the calculus comprises a factoring rule and, in the presence of equality, the usual superposition rules as well as superposition into the constraint. We skip those rules here because they are not relevant for the remainder of this article. For full details cf. [16].

This inference system is sound and complete as a slight variation of [15, 14], provided that constraint satisfiability is decidable. As an example, this is the case when the constraints are increasing (cf. Theorem 11 below).

We now extend the calculus to an inference system including a *melting* rule [16], which serves as a limited form of induction. To define melting, we need the notion of an ancestor. In any of the usual inferences, the *ancestors* of the conclusion are the rightmost premise and all of its ancestors.

Melting:

$$\frac{(\vec{x}\approx\vec{y})\bar{\sigma} \parallel C \quad (\vec{x}\approx\vec{y})\bar{\sigma}\bar{\tau}' \parallel C'}{(\vec{x}\approx\vec{y})\bar{\sigma}'' \parallel C}$$

where (1) $(\vec{x}\approx\vec{y})\bar{\sigma} \parallel C$ is an ancestor of $(\vec{x}\approx\vec{y})\bar{\sigma}\bar{\tau}' \parallel C'$, and (2) $(\vec{x}\approx\vec{y})\bar{\sigma}\bar{\tau}' \parallel C'$ is a variant of $(\vec{x}\approx\vec{y})\bar{\sigma}\bar{\tau} \parallel C$, and either (3.i) $\bar{\sigma}$ is of the form $\bar{\sigma} = \bar{\sigma}_1\bar{\sigma}_2^*$ and $\bar{\sigma}'' = \bar{\sigma}_1(\bar{\sigma}_2|\bar{\tau})^*$, or (3.ii) $\bar{\sigma}$ is not of this form and $\bar{\sigma}'' = \bar{\sigma}\bar{\tau}^*$.

The ancestors of the conclusion of a melting inference are defined as the ancestors of the leftmost premise.

4.3 Correctness

Intuitively, the melting rule states: if it is possible to derive $\alpha\bar{\tau} \parallel C$ from $\alpha \parallel C$, then it is also possible to repeat this process to derive $\alpha\bar{\tau}\bar{\tau} \parallel C$ and so on. This is of course not always true, i.e. melting-like rules tend to be incorrect.

Example 9 *Let P denote the at-most-one predicate over the natural numbers, encoded by the following set of constrained clauses:*

$$N = \{(x{\approx}y)\{y \mapsto 0\} \parallel P(x), \quad (x{\approx}y)\{y \mapsto s(y)\}\{y \mapsto 0\} \parallel P(x)\} \ .$$

This is of course equivalent to just $\{P(0),\ P(1)\}$. A melting derivation of the form

$$\frac{(\vec{x}{\approx}\vec{y})\bar{\sigma} \parallel P(x) \quad (\vec{x}{\approx}\vec{y})\bar{\sigma}\bar{\tau}' \parallel P(x)}{(x{\approx}y)\{y \mapsto s(y)\}^*\{y \mapsto 0\} \parallel P(x)}$$

would derive a constrained clause that is equivalent to $P(x)$, and thus would certainly not be sound! This is why melting contains the restriction that one premise must be an ancestor of the other one.

Even with this restriction, melting could still yield incorrect results. Consider the following set of constrained clauses:

$$N = \{(x{\approx}y)\{y \mapsto 0\} \parallel P(x), \quad (x{\approx}y)\{y \mapsto 0\} \parallel P(x) \to P(s(x))\} \ ,$$

i.e. $P(0)$ holds, and $P(0)$ implies $P(s(0))$. We can easily derive $P(1)$, which brings us back into the situation of the first part of this example, except that the second premise is now derived from the first one. Melting is still unsound.

The correctness of melting relies on the derivation from one premise to the other being repeatable. This does hold in all examples in this paper, and it is also guaranteed whenever there are no inferences (except for melting inferences) where both premises are constrained. The latter can for instance be ensured by considering only Horn clauses where all clauses with a positive literal are unconstrained; cf. [16, 12] for more details.

4.4 Termination

Melting is important because it allows us in many cases to saturate clause sets much faster, or even to turn an infinite saturation into a finite one.

Example 10 *Consider the clause set $\{x{\approx}y \parallel P(x) \to Q(y),\ P(x) \to P(s(x))\}$, where the ordering is chosen such that $P(t_1) \succ Q(t_2)$ for all ground terms t_1, t_2. When saturating this clause set, we successively derive all clauses of the form $s^n(x){\approx}y \parallel P(x) \to Q(y)$ by iterating the following inference step:*

$$\frac{P(x) \to P(s(x)) \quad s^n(x){\approx}y \parallel P(x) \to Q(y)}{s^{n+1}(x){\approx}y \parallel P(x) \to Q(y)}$$

This derivation does not terminate. With melting however, we can make make one such inference to derive $s(x)\approx y \parallel P(x) \to Q(y)$ *and directly follow up with a melting inference:*

$$\frac{x\approx y \parallel P(x) \to Q(y) \quad s(x)\approx y \parallel P(x) \to Q(y)}{s^*(x)\approx y \parallel P(x) \to Q(y)}$$

(where $s^*(x)\approx y$ *stands for* $(x\approx y)\{x \mapsto s(x), y \mapsto y\}^*$*). After this inference, the clause set is already saturated.*

The ground instances of constrained clauses that are derivable with and without melting coincide. However, note that **Melting** does not make the inference system terminating in general. In [16], we presented conditions under which termination results for saturation on unconstrained nonequational clauses carry over to constrained clauses. All examples in this paper lie in such fragments. Extending these results to clauses containing equational literals will be the subject of further work.

5 Locality and Melting Constraints

In many settings, the regular constraints that appear just stack increments on both sides of an equation. This is for example the case with the strict monotonicity[4] axiom $s(x)\approx y \to s(f(x)) \leq f(y)$, which gives rise to clauses of the form $s^n(x)\approx y \to s^n(f(x)) \leq f(y)$ for each $n \geq 0$, or to the constrained clause

$$(x\approx y, v\approx w)\sigma^* \parallel v\approx f(x) \to w \leq f(y)$$

for $\sigma = \{x \mapsto s(x), y \mapsto y, v \mapsto s(v), w \mapsto w\}$. If $f : S_1 \to S_2$ is a function symbol that connects two different sorts (which have different successor functions s_1, s_2), then σ would be $\sigma = \{x \mapsto s_1(x), y \mapsto y, v \mapsto s_2(v), w \mapsto w\}$.

Let $\bar{\sigma}$ be a substitution expression over Σ. Let Σ contain for each sort S of a domain element of $\bar{\sigma}$ a unique unary function symbol $s_S : S \to S$. Then $\bar{\sigma}$ is *increasing for* Σ if, for each basic substitution τ in σ and each variable x, either $\tau(x)$ is a constant or $\tau(x) = s_S^k(x)$ for some $k \geq 0$, where S is the sort of x. A regular constraint is *increasing for* Σ if it is empty or if its substitution expression is of the form $\bar{\sigma}_1 \circ \cdots \circ \bar{\sigma}_n$ such that each $\bar{\sigma}_i$ is increasing for some $\Sigma_i \subseteq \Sigma$, and $\Sigma_i \cap \Sigma_j$ contains only constants for $i \neq j$.

We use a shorthand notation for increasing substitution expressions and increasing constraints: For example, if $\sigma = \{x \mapsto s(s(x)), y \mapsto y\}$ and $\tau = \{x \mapsto 0, y \mapsto 0\}$, we write $\sigma^*\tau$ as $(s^2, s^0)^*(0, 0)$, and we write the constraint $(x\approx y)\sigma^*\tau$ as $(x\approx y)(s^2, s^0)^*(0, 0)$. With this notation, the constrained clause describing strict monotonicity becomes $(x\approx y, v\approx w)(s^1, s^0, s^1, s^0)^* \parallel v\approx f(x) \to w \leq f(y)$, or

$$(x\approx y, v\approx w)(s_1^1, s_1^0, s_2^1, s_2^0)^* \parallel v\approx f(x) \to w \leq f(y)$$

[4] Similar considerations also hold for the monotonicity axiom $s(x)\approx y \to f(x) \leq f(y)$.

Table 1. Simplification rules for increasing substitutions

$(s^k, z) \rightsquigarrow (s^k, s^0)(s^0, z)$	where z is a constant and $k > 0$
$(z, s^k) \rightsquigarrow (s^0, s^k)(z, s^0)$	where z is a constant and $k > 0$
$(z, s^l)\bar{\sigma} \rightsquigarrow \bar{\sigma}'(z, s^l)$	where z is a constant, $\lvert \mathrm{dom}(\bar{\sigma})\rvert = 1$, and $\bar{\sigma}'$ arises by padding each basic substitution τ in $\bar{\sigma}$ with s^0, forming (s^0, τ)
$(s^l, z)\bar{\sigma} \rightsquigarrow \bar{\sigma}'(s^l, z)$	similarly
$(s^0, s^0) \circ \bar{\sigma} \rightsquigarrow \bar{\sigma}$	
$(s^0, s^0)^* \rightsquigarrow (s^0, s^0)$	
$(\bar{\sigma}^*)^* \rightsquigarrow \bar{\sigma}^*$	
$(\bar{\sigma}^*\bar{\tau}^*)^* \rightsquigarrow \bar{\sigma}^*\bar{\tau}^*$	
$(\bar{\sigma}\bar{\tau}^*)^* \rightsquigarrow (s^0, s^0) \mid \bar{\sigma}\bar{\sigma}^*\bar{\tau}^*$	if $\bar{\sigma}$ is not of the form $\bar{\rho}^*$
$(\bar{\sigma} \mid \bar{\tau})^* \rightsquigarrow \bar{\sigma}^*\bar{\tau}^*$	
$(s^{k+1}, s^{l+1}) \rightsquigarrow (s^k, s^l)$	
$(s^k, s^l)(s^m, s^n) \rightsquigarrow (s^{k+m}, s^{l+n})$	
$(s^k, s^0)^*(s^0, s^l)^* \rightsquigarrow (s^g, s^0)^* \mid (s^0, s^g)^*$	where $g = \gcd(k, l)$
$(s^k, s^0)^*(s^l, s^0)^* \rightsquigarrow \Big((s^0, s^0) \mid (s^k, s^0) \mid \ldots \mid (s^{k\cdot(l-1)}, s^0)\Big)\,(s^l, s^0)^*$	

if the domain and range of f have different sorts and we have two successor functions. If in addition the substitution operates independently on the different variables of the constraint, we simplify the notation even more and write, for example, $x{\approx}s^*(s(y))$ for $(x{\approx}y)\{x \mapsto x, y \mapsto s(y)\}\{x \mapsto x, y \mapsto s(y)\}^*$.

Increasing constraints are one class where the constrained calculus is applicable:

Theorem 11 *Let Σ be a set of function symbols and α an increasing regular constraint for Σ. The satisfiability of α is decidable.*

Proof. The proof proceeds by rewriting the substitution expression of α until the only remaining loops are of the form $(s_1^k, s_2^0, \ldots, s_n^0)^*$ and loops over the same unary function symbols cannot interact. We make heavy use of the fact that α is interpreted syntactically, which means that we can consider the appearing function symbols as free constructors.

The regular constraint α's substitution expression is by definition of the form $\bar{\sigma}_\alpha = \bar{\sigma}_1 \circ \cdots \circ \bar{\sigma}_n$ such that each $\bar{\sigma}_i$ is increasing for $\Sigma_i \subset \Sigma$.

For now let us concentrate on a single increasing substitution expression $\bar{\sigma}_i$ for Σ_i. Note that $\circ$ is not only associative but also commutative on expressions with the same domain in this setting. Simplify the substitution expression using the rewrite rules from Table 1. (To improve readability, in Table 1 we only give the rules for a domain of size 2 and, assuming that both domain elements have the same type, a single function symbol $s \in \Sigma_i$; larger domains can be handled in a similar way.)

After each rewrite step, the whole substitution expression is additionally normalized with respect to the distribution rule $\bar{\sigma}(\bar{\tau}_1 \mid \bar{\tau}_2) \rightsquigarrow \bar{\sigma}\bar{\tau}_1 \mid \bar{\sigma}\bar{\tau}_2$.

Correctness is obvious for most rules. The next to last rule follows directly from Bézout's formula, which states that for any integers k and l there exist integers m and n such that $km+ln = \mathsf{gcd}(k,l)$ and that for all integers k, l, m, n, the number $km + ln$ is a multiple of the greatest common divisor of k and l (negative multiples of the gcd can also be reached). The last rule follows from $kl = lk$: each set of l unfoldings of the first loop $(s^k, s^0)^*$ is equivalent to some unfoldings of the second loop $(s^l, s^0)^*$.

This rewrite system terminates because in each step the quintuple of

(1) the number of basic substitutions that add at least one s to one variable and ground another variable,
(2) the number of basic substitutions with a domain size < 2,
(3) the multiset of the star depths (i.e. the number of $*$ operators above) of all appearing substitution expressions,
(4) the number of basic substitutions, and
(5) the number of occurrences of s

decreases lexicographically.

The individual rules are reducing due to elements 1, 1, 2, 2, 3-4, 3, 3, 3, 3, 3, 5, 3-4, 5, and 3 of this tuple, respectively. Duplicating rules like

$$(\bar{\sigma}\bar{\tau}^*)^* \rightsquigarrow (s^0, s^0) \mid \bar{\sigma}\bar{\sigma}^*\bar{\tau}^*$$

cannot increase 1 or 2 because they are only applied once there are no more mixed increasing/grounding and no small-domain substitutions left.

When no more rules are applicable, the substitution expression contains at most one star in each disjunct, in a subexpression of the form $(s^k, s^0)^*$ or $(s^0, s^k)^*$.[5] Checking satisfiability of such constraints is easy.

Larger domains are handled by straightforward adaptions of the rules on basic substitutions, such as

$$(s_1^k, s_2^l, z) \rightsquigarrow (s_1^k, s_2^l, s_3^0)(s_1^0, s_2^0, z) \text{ where } z \text{ is a constant, and } k > 0 \text{ or } l > 0$$

for the first rule. These rules rely on the fact that there is only one function symbol per sort, i.e. different function symbols cannot interact.
If the regular constraint's substitution expression has more than one increasing component, then the above rules are applied to each component independently: Since the different components can only add different unary function symbols, the only interaction between the components comes from (partially) grounding basic substitutions: Remember that (s_1^0, z) and (s_2^0, z) describe the same basic substitution, so grounding substitutions can be regarded as a part of any component, and they are successively moved all the way up in the hierarchy. □

Example 12 *Examples of clause sets where the use of constraints comes in handy:*

[5] Note that $\bar{\sigma}_\alpha$ is well-defined by definition of α and the rewrite rules respect well-formation, so for example no grounding substitutions can appear below a $*$, and a subexpression of the form $(z_1, z_2)\bar{\sigma}$ also cannot appear.

- *Let all function symbols s_i be unary. For subterm locality, $N[\preceq G]$ is equivalent to the set of clauses of the form*

$$(s_1|\ldots|s_n)^*(x_1)\approx t_1, \ldots, (s_1|\ldots|s_n)^*(x_m)\approx t_m, \alpha \parallel C \ ,$$

  where $\alpha \parallel C \in N$ has the free variables $x_1, \ldots, x_m$ and $t_1, \ldots, t_m$ are maximal terms in G (w.r.t. the subterm ordering). That will be much more concise than writing down all of $N[\preceq G]$ because the number of instances goes down from $|\mathsf{st}(G)|^m$ to $|\mathsf{maxst}(G)|^m$ ($\mathsf{maxst}(G)$ are the maximal subterms of G).
- *The standard ordering over the naturals is completely described by the saturated set $\{y\approx s^+(x) \parallel x < y,\ y\approx s^+(x) \parallel y \not< x\}$.*
- *Cycle-free lists can be characterized without a reachability predicate just by the clause $y\approx p^*(p(x)) \parallel x\approx y \rightarrow x\approx nil$. More generally, a similar clause characterizes absolutely free unary constructors.*

As mentioned before, Theorem 7, a result by Basin and Ganzinger [5], does not hold for full superposition instead of ordered resolution. Theorem 13 below shows how a slightly restricted form of superposition allows to recover locality, and that this is even true for clauses with regular constraints. Our proof extends the one from [5].

In the following, saturation can always be interpreted as saturation with or without melting, because (due to the definition of ground clauses) the constraints do not affect the proofs.

Theorem 13 *Let $\prec_\mathcal{T}$ be a reduction ordering and $\prec$ a compatible and total atom ordering. Let N be a set of constrained clauses that is reductive w.r.t. $\prec_\mathcal{T}$ and saturated w.r.t. $\prec$. Let each constrained clause in N with positive equational atoms contain either a unique positive equation which is also maximal, or a negative equation which is maximal. Let C be a ground clause whose succedent does not contain any equations. Then $N \models C$ iff $N \models_\preceq C$.*

Proof. The direction $N \models_\preceq C \implies N \models C$ is obvious and independent of the properties of N and C.

For the opposite direction, assume that N and C satisfy the preconditions of the theorem, and that $N \models C$, i.e. $N \cup \neg C$ is unsatisfiable. Because N is saturated, a set of support strategy for N is complete [3], i.e. there is a proof of the unsatisfiability of $N \cup \neg C$ in which at most one premise of every inference is an instance of a clause in N.

We prove by induction on the length of this proof that

- all clauses that are used in the proof are $\preceq C$,
- all clauses that are derived in the proof contain equational atoms only negatively, and
- all clauses that are derived in the proof are unconstrained ground clauses.

As the base case, consider an inference between input clauses. It cannot be a factoring or equality factoring inference because N is saturated and because all

elements of $\neg C$ are unit clauses. It also cannot be a melting inference for the same reason. So it must be an inference by ordered resolution or by superposition.

If it is an inference by ordered resolution, assume that the first premise is an instance $\alpha \parallel \Gamma \to \Delta, A$ of a clause in N and the second premise is a clause $A \to$ from $\neg C$. Since A is ground and maximal in $\Gamma \to \Delta, A$ and α cannot contain any extra variables, the whole premise is ground and $\preceq C$, and so is the conclusion $\alpha \parallel \Gamma \to \Delta$ of the inference. The constraint α must be valid since $\alpha \parallel \Gamma \to \Delta, A$ is ground and not redundant, so the conclusion is unconstrained. Since Δ does not contain any equations, the conclusion does not contain any positive equations. The situation is similar if the second premise is an instance of a clause from N, and the case where both premises come from $\neg C$ is trivial. If the inference is a superposition inference, only the first premise must stem from N. Let this premise be $\alpha \parallel \Gamma \to \Delta, s \approx t$ with s being maximal. Similarly to above, it follows that s is ground and $s \approx t \preceq C$, and hence all of the premise is ground and $\preceq C$ and the same holds for the conclusion. Again Δ does not contain any equations and so the conclusion does not contain any positive equations.

For the induction step, consider a later inference in the proof. Again, it cannot be a melting inference because the induction hypothesis states that all previously derived clauses are unconstrained. It also cannot be an equality factoring inference because derived clauses do not contain positive equations.

If it is a factoring or equality resolution inference, the premise must be a derived, and hence ground, clause. Since the inference does not introduce any new literals, it is obvious that the properties to be proven are passed on from the premise to the conclusion. The situation is similar for an inference by ordered resolution where both premises are from $\neg C$ or derived. Superposition inferences from two clauses outside N are not possible because, by the conditions on C and the induction hypothesis, none of these clauses contain positive equations.

For an ordered resolution or superposition inference where one premise is an instance of a clause in N (for superposition, this must be the first premise), the reasoning is similar to the reasoning in the base case, because the partner premise is ground and cannot contribute any positive equations by induction hypothesis. □

In some cases, saturation is still too strong a requirement to arrive at finite local axiomatizations.

Example 14 *Consider the theory of a strictly monotone function f as described in the beginning of this section, together with the theory of preorders:*

$$N_{\leq,f} = \{x \leq x, \quad x \leq y \wedge y \leq z \to x \leq z, \quad \alpha \parallel v \approx f(x) \to w \leq f(y)\}$$

For now, we ignore the exact shape of the constraint α. The axiomatization $N_{\leq,f}$ is local, but we cannot show this using the previous theorem, because inferences between the clauses for transitivity and monotonicity allow the derivation of additional clauses of the form

$$\alpha \parallel v \approx f(x), w_1 \leq w_2 \leq \ldots \leq w \to w_1 \leq f(y) \ .$$

When the theory does not contain equality literals, this can be remedied by considering a straightforward extension of peak saturation to constrained clauses:

An inference $\alpha_1 \parallel C_1,\ \alpha_2 \parallel C_2 \vdash \alpha \parallel C$ between constrained clauses is called a *peak inference* (resp. *plateau inference*) if the unconstrained inference $C_1, C_2 \vdash C$ is a peak inference (resp. plateau inference). A set of constrained Horn clauses N is *peak saturated* if all peak inferences are redundant for which (i) the second, negative premise is in N and (ii) the first, positive premise's antecedent does not contain a maximal term, and (iii) the positive premise is in N or generated from N using plateau inferences.

Theorem 15 *Let $\prec_{\mathcal{T}}$ be a well-founded term ordering and $\prec$ a compatible and total atom ordering. Let N be a set of constrained Horn clauses without equality that is reductive w.r.t. $\prec_{\mathcal{T}}$ and peak saturated w.r.t. $\prec$. Let C be a ground clause without equality. Then $N \models C$ iff $N \models_{\preceq} C$.*

Proof. The proof of the respective theorem for unconstrained clauses in [5] only makes use of properties of ground inferences. Because the ground instances of constrained clauses are defined in such a way that they are unconstrained, that proof carries over to constrained clause sets word by word. Note that for constrained clauses without equality literals, the only applicable rules are ordered resolution and factoring. □

Example 16 *In our running example of strict monotonicity, the only nonredundant inference that is enabled for $N_{\leq,f}$ is between the constrained clause for monotonicity and an instance of transitivity:*

$$\frac{\alpha \parallel v \approx f(x) \to w \leq f(y) \quad \parallel w' \leq w \wedge w \leq f(y) \to w' \leq f(y)}{\alpha \parallel v \approx f(x), w' \leq w \to w' \leq f(y)}$$

(There is also an inference into the other antecedent literal of the transitivity clause, for which the situation is similar.) It is a plateau inference because $f(y)$ appears in the succedent of the right premise. The inferences from this clause and transitivity are again plateau inferences, and the same holds for all further inferences. Since no peak inferences are possible, $N_{\leq,f}$ is peak-saturated. By Theorem 15, $N_{\leq,f}$ is also order local.

Example 17 *Other examples where the theorem directly proves locality because the constrained clause sets are (peak) saturated:*

- *The theory axiomatized by:* $\{\ x \approx s^*(0) \parallel P(x)\}$
- *The theory of even and odd numbers axiomatized as follows:*

$$\{\ x \approx (ss)^*(0) \parallel \mathit{Even}(x), \quad x \approx (ss)^*(s(0)) \parallel \mathit{Odd}(x)\ \}$$

- *The theory of number pairs with an even sum (cf. [5]):*

$$\begin{aligned}\{\ & x \approx (ss)^*(0) \quad , y \approx (ss)^*(0) \quad \parallel \mathit{Evensum}(x, y),\\ & x \approx (ss)^*(s(0)), y \approx (ss)^*(s(0)) \parallel \mathit{Evensum}(x, y)\ \}\end{aligned}$$

– *Monotonicity on intervals* $\{a_i, \ldots, b_i\}$, $i \in I$, *where the* a_i *and* b_i *are concrete natural numbers:*

$$N_{\leq} \cup \bigcup_{i \in I} \{\ s^*(a_i) \approx x, s^*(x) \approx y, s^*(y) \approx b_i \parallel f(x) \leq f(y)\ \}$$

– *Monotonicity on unbounded intervals like* $\{a, \ldots\}$ *and* $\{\ldots, b\}$ *(similar).*

Example 18 *Consider the theory axiomatized by* $N = N_0 \cup \{\ x \approx s^*(0) \parallel P(x)\}$, *where* N_0 *is the axiomatization of natural numbers consisting of the absolutely free constructor axioms for* s *and* 0 *(a set of Horn clauses which is local [25] and can be proved to be order local also using Theorem 15). Let* $C = P(a)$, *where* a *is a new constant such that* $0 \prec a \prec s(0)$. *By locality, we know that* $N \models C$ *iff* $N[\preceq C] \models C$. *Then* $N[\preceq C] = \{P(0)\}$, *so* $N[\preceq C] \cup \neg C$ *is satisfiable. This shows that* $N \not\models C$. *(This is explained by the fact that we do not consider satisfiability in the initial model of the natural numbers, but in* some *model of* N.*)*

Often the axiomatizations have the form $N = N_0 \cup \mathcal{K}$, where N_0 is an axiomatization of a base theory and $\mathcal{K}$ defines properties of additional functions. The constraints in the saturated form may suggest ways of defining new predicates which would allow to use a finite clause notation. Assume f has sort $i \to o$. The clause $s^*(x) \approx y \parallel f(x) \leq f(y)$ can be replaced for instance with $x \leq' y \to f(x) \leq f(y)$, where $\leq'$ is a new predicate. The form of the constraint guides in giving a (possibly recursive) definition for $\leq'$. In future work we will analyze such situations, as well as possibilities of checking satisfiability in the initial model.

5.1 Extended Example: Monotonicity

The following example illustrates the whole workflow of the melting-based approach to locality. Consider the following theory of monotonicity:

$$N_1 = \mathsf{Pre} \cup \{f(x) \leq f(s(x))\}\ .$$

This theory is not local. For example, the weak partial model $\mathcal{A}$ over the natural numbers with $f^{\mathcal{A}}(0) = 1$ and $f^{\mathcal{A}}(s(s(0))) = 0$ cannot be embedded into a total model of N_1.

Using either full saturation or peak saturation (without melting), we arrive in the limit at a clause set containing

$$N_2 = N_{\leq} \cup \{f(x) \leq f(s^n(x)) \mid n \in \mathbb{N}\}\ .$$

This set is local by Theorem 7. Unfortunately, it is also infinite.

With melting, however, we arrive in a finite number of steps at a saturated set containing

$$N_3 = N_{\leq} \cup \{y \approx s^*(s(x)) \parallel f(x) \leq f(y)\}\ .$$

This set is local by Theorem 13. It is finite, so we can use it to decide queries on N_1.

Assume that we want to decide whether or not $N_1 \models \{f(0) \leq f(s(0))\}$. Equivalently, we want to decide whether or not $N_1 \cup G$ is unsatisfiable, where $G = \{f(0) \leq f(s(s(0))) \to \bot\}$. We have now two ways to decide this, using that N_1 and N_3 are equivalent and N_3 is local:

Approach 1: We can use locality of N_3 to decide whether $N_3[\leq G] \cup G$ is unsatisfiable. This is indeed the case:

$$\frac{f(0) \leq f(s(0)) \quad f(0) \leq f(s(0)) \to}{\bot}$$

The leftmost premise is an element of $N_3[\leq G]$, in particular it is a ground instance of $y \approx s^*(s(x)) \parallel f(x) \leq f(y)$, where $x \mapsto 0$ and $y \mapsto s(0)$.

Approach 2: Because saturation of $N_3 \cup G$ terminates, we can alternatively use melting-based saturation again to decide whether $N_3 \cup G$ is unsatisfiable. This is verified by the following one-step derivation:

$$\frac{y \approx s^*(s(x)) \parallel f(x) \leq f(y) \quad f(0) \leq f(s(0)) \to}{s(0) \approx s^*(s(0)) \parallel \bot}$$

and the final constraint is satisfiable.

The first approach is applicable for any ground G. The second one is applicable to non-ground G without Skolem constants, provided all variables are universally quantified. The treatment of queries only existential variables or with a single $\exists\forall$ alternation is also possible, but it requires more elaborate constraints (cf. [16] for details).

5.2 Limitations

For computing finite saturations (or peak saturations) of sets of constrained clauses it is very important to have a decision procedure for checking satisfiability of the constraints. In Theorem 11 we showed that if the constraints only contain unary constructors, then checking satisfiability is decidable. However, satisfiability is undecidable for most extensions of the presented fragment of regular constraints. This be proven by a reduction of the Post correspondence problem.

Theorem 19 *Satisfiability of regular constraints with substitution expression σ is undecidable, even if all function symbols are at most unary and σ satisfies all conditions of an increasing substitution expression except the first.*

Proof. We show that with every Post correspondence problem P we can associate a regular constraint α with substitution expression σ (which satisfies all conditions of an increasing substitution expression except the first) such that P has a solution if, and only if, α is satisfiable:

Consider a Post correspondence problem over the alphabet $\{a, b\}$ with given word pairs $(u_1, v_1), \ldots, (u_n, v_n)$. We model words by ground terms over the unary function symbols a, b with empty word 0.

Let $\sigma_a(x,y) = (a(x), a(y))$, $\sigma_b(x,y) = (b(x), b(y))$, and $\sigma_0(x,y) = (0,0)$. Then the following expression denotes all nonempty pairs of equal words:

$$(x,y)(\sigma_a|\sigma_b)^*\sigma_0$$

Moreover let $\sigma_i(x,y) = (u_i(x), v_i(y))$ for $i = 1, \ldots, n$. Then the following expression denotes all word pairs that can be formed from the (u_i, v_i):

$$(x,y)(\sigma_1|\ldots|\sigma_n)^*\sigma_0$$

Finally, the PCP has a solution if, and only if, the following constraint is satisfiable:

$$(x,y)(\sigma_a|\sigma_b)^*\sigma_0 \approx (x,y)(\sigma_1|\ldots|\sigma_n)^*\sigma_0 \ . \qquad \square$$

6 Conclusion

In this paper we presented a method for obtaining finite representations of local sets of clauses from possibly non-local ones. We extended the work of Basin and Ganzinger [4, 5]: In order to address the fact that saturation can generate infinite sets of clauses, we used constrained clauses, which allow us to give a finite representation for possibly infinite saturated sets of clauses and defined an ordered resolution and superposition calculus for such constrained clauses. We established links between locality and saturation for constrained clauses. The form of the constraints suggests definitions for new predicates which would yield saturated – hence local – theory axiomatizations.

In future work we would like to analyze possibilities of reasoning in the initial model (not investigated here), and ways of defining new predicate symbols, e.g. using recursive definitions; we hope that some of the results in [25] could prove useful for this. We plan to study the applicability of our results to reasoning in various data structures.

Acknowledgments. This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS). See `www.avacs.org` for more information.

References

[1] A. Armando, M. P. Bonacina, S. Ranise, and S. Schulz. New results on rewrite-based satisfiability procedures. *ACM Trans. Comput. Log.*, 10(1), 2009.

[2] A. Armando, S. Ranise, and M. Rusinowitch. A rewriting approach to satisfiability procedures. *Inf. Comput.*, 183(2):140–164, 2003.

[3] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. of Logic and Computation*, 4(3):217–247, 1994.

[4] D. Basin and H. Ganzinger. Complexity analysis based on ordered resolution. In *Proc. 11th IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 456–465. IEEE Computer Society Press, 1996.

[5] D. A. Basin and H. Ganzinger. Automated complexity analysis based on ordered resolution. *Journal of the ACM*, 48(1):70–109, 2001.
[6] W.F. Dowling and J.H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming*, 1(3):267–284, 1984.
[7] H. Ganzinger. Relating semantic and proof-theoretic concepts for polynomial time decidability of uniform word problems. In *Proc. 16th IEEE Symposium on Logic in Computer Science (LICS'01)*, pages 81–92. IEEE Computer Society Press, 2001.
[8] H. Ganzinger, V. Sofronie-Stokkermans, and U. Waldmann. Modular proof systems for partial functions with Evans equality. *Information and Computation*, 204(10):1453–1492, 2006.
[9] R. Givan and D. McAllester. New results on local inference relations. In *Principles of Knowledge Representation and reasoning: Proceedings of the Third International Conference (KR'92)*, pages 403–412. Morgan Kaufmann Press, 1992.
[10] R. Givan and D.A. McAllester. Polynomial-time computation via local inference relations. *ACM Transactions on Computational Logic*, 3(4):521–541, 2002.
[11] D. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2):284–303, 1993.
[12] M. Horbach. *Superposition-based Decision Procedures for Fixed Domain and Minimal Model Semantics*. PhD thesis, Max Planck Institute for Computer Science and Saarland University, 2010.
[13] M. Horbach and V. Sofronie-Stokkermans. Obtaining finite local theory axiomatizations via saturation. In P. Fontaine and R. Schmidt, editors, *Proc. of the 9th International Symposium on Frontiers of Combining Systems, FroCoS 2013*, *LNAI*. Springer, 2013. to appear
[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 08*, *LNCS 5213*, pages 293–307. Springer, 2008.
[15] M. Horbach and C. Weidenbach. Decidability results for saturation-based model building. In R. Schmidt, editor, *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction*, *LNAI 5663*, pages 404–420. Springer, 2009.
[16] M. Horbach and C. Weidenbach. Deciding the inductive validity of $\forall\exists^*$ queries. In E. Grädel and R. Kahle, editors, *Proc. of the 18th Annual Conference of the European Association for Computer Science Logic, CSL 2009*, *LNCS 5771*, pages 332–347. Springer, 2009.
[17] C. Ihlemann, S. Jacobs, and V. Sofronie-Stokkermans. On local reasoning in verification. In C. R. Ramakrishnan and J. Rehof, editors, *Proc. 14th International Conference, TACAS 2008*, *LNCS 4963*, pages 265–281. Springer, 2008.
[18] C. Ihlemann and V. Sofronie-Stokkermans. On hierarchical reasoning in combinations of theories. In J. Giesl and R. Hähnle, editors, *Automated Reasoning, 5th International Joint Conference, IJCAR 2010*, *LNAI 6173*, pages 30–45. Springer, 2010.
[19] H. Kirchner, S. Ranise, C. Ringeissen, and D.-K. Tran. On superposition-based satisfiability procedures and their combination. In D. V. Hung and M. Wirsing, editors, *ICTAC 2005, Second International Colloquium on Theoretical Aspects of Computing*, *LNCS 3722*, pages 594–608. Springer, 2005.
[20] C. Lynch and B. Morawska. Automatic decidability. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002)*, pages 7–. IEEE Comp. Soc., 2002.
[21] C. Lynch, S. Ranise, C. Ringeissen, and D.-K. Tran. Automatic decidability and combinability. *Inf. Comput.*, 209(7):1026–1047, 2011.

[22] R. Nieuwenhuis and A. Rubio. Theorem proving with ordering constrained clauses. In D. Kapur, editor, *Automated Deduction - CADE-11, 11th International Conference on Automated Deduction*, *LNCS 607*, pages 477–491. Springer, 1992.

[23] T. Skolem. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit und Beweisbarkeit mathematischen Sätze nebst einem Theoreme über dichte Mengen. *Skrifter utgit av Videnskabsselskapet i Kristiania, I. Matematisk-naturvidenskabelig klasse, 4*, pages 1–36, 1920.

[24] V. Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. In R. Nieuwenhuis, editor, *Automated Deduction - CADE-20, 20th International Conference on Automated Deduction, LNAI 3632*, pages 219–234. Springer, 2005.

[25] V. Sofronie-Stokkermans. Locality results for certain extensions of theories with bridging functions. In R. A. Schmidt, editor, *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction*, *LNAI 5663*, pages 67–083. Springer, 2009.

[26] E. Tushkanova, C. Ringeissen, A. Giorgetti, and O. Kouchnarenko. Automatic decidability: A schematic calculus for theories with counting operators. *Proc. 24th International Conference on Rewriting Techniques and Applications (RTA 2013), LIPIcs 21*, pages 303-318, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

[27] 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.