Decision Procedures for Automating Termination Proofs

Ruzica Piskac[1] and Thomas Wies[2]

[1] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[2] Institute of Science and Technology (IST), Austria

Abstract. Automated termination provers often use the following schema to prove that a program terminates: construct a relational abstraction of the program's transition relation and then show that the relational abstraction is well-founded. The focus of current tools has been on developing sophisticated techniques for constructing the abstractions while relying on known decidable logics (such as linear arithmetic) to express them. We believe we can significantly increase the class of programs that are amenable to automated termination proofs by identifying more expressive decidable logics for reasoning about well-founded relations. We therefore present a new decision procedure for reasoning about multiset orderings, which are among the most powerful orderings used to prove termination. We show that, using our decision procedure, one can automatically prove termination of natural abstractions of programs.

1 Introduction

The standard technique for proving program termination is to construct a *ranking function* [11]. A ranking function maps the states of the program into some well-founded domain, i.e., a set equipped with a well-founded ordering. The mapping is such that, with each transition taken by the program, the value of the ranking function decreases in the ordering. The canonical well-founded ordering for constructing ranking functions is the strict order on the natural numbers. However, constructing *global* ranking functions for this ordering (i.e., functions that decrease with every transition of the program) requires a lot of ingenuity.

Despite the general result of undecidability of the halting problem, recent advances in program analysis have brought forth tools that can automatically prove termination of real-world programs [3, 5, 6]. The success of these tools is due to the development of new proof techniques for termination [15, 23]. These techniques avoid the construction of a global termination argument and, instead, decompose the program into simpler ones. Each of these simpler programs is then proved terminating independently, by constructing a simpler ranking function. The automation of these proof techniques relies on decision procedures for reasoning about constraints on well-founded domains. The existing tools use known decidable logics such as linear arithmetic to express these constraints [4, 22], which effectively restricts the range of ranking functions that can be constructed automatically. We believe that by providing decision procedures for more sophisticated well-founded domains, one can significantly increase the class of programs that are amenable to automated termination proofs.

Among the most powerful well-founded domains for proving program termination are multiset orderings [9]. In this paper, we present a decision procedure for automated reasoning about such orderings. A multiset is an unordered collection of elements from a base set, where each element may occur multiple times. For instance, $\{1, 2, 2, 3\}$ is a multiset of natural numbers. This multiset is equal to the multiset $\{1, 3, 2, 2\}$ but different from the multiset $\{1, 2, 3\}$. A (strict) ordering $\prec$ on the base set S can be lifted to an ordering $\prec_{\mathsf{m}}$ on (finite) multisets over S as follows. For two multisets X and Y, $X \prec_{\mathsf{m}} Y$ holds iff X and Y are different and for every element $x \in S$ which occurs more times in X than in Y, there exists an element $y \in S$ which occurs more times in Y than in X and $x \prec y$. For instance, $\{1, 1, 1, 2, 2\} \prec_{\mathsf{m}} \{1, 3\}$ since $1 < 3$ and $2 < 3$. Multiset orderings are interesting because they inherit important properties of the ordering on the base set. In particular, the multiset ordering $\prec_{\mathsf{m}}$ is well-founded iff the ordering $\prec$ on the base set is well-founded [9]. Multiset orderings have been traditionally used for manual termination proofs in program verification [7, 9], term rewriting systems [1, 8], and theorem proving [2, 16]. The question whether reasoning about multiset orderings can be effectively automated was open.

Contributions. In this paper, we present a new logic called POSSUM for expressing ordering constraints on finite multisets. The logic is parameterized by the theory of the base set, which can be an arbitrary theory equipped with a preorder (not necessarily well-founded). We show that if the base theory is decidable then so is its extension to a multiset ordering. What is more, we show that if the base theory is decidable in NP then the satisfiability problem for its POSSUM extension is NP-complete. Our decision procedure can be easily implemented using off-the-shelf SMT solvers. We demonstrate the usefulness of our decision procedure for proving termination of interesting programs. We therefore believe that it can be a useful component of future automated termination provers.

2 Examples

We motivate the usefulness of our decision procedure for proving termination through two examples.

Example 1: counting leaves in a tree. Our first example is a program taken from [9] and shown in Figure 1. The termination behavior of this program is representative for many programs that traverse algebraic data types.

The program CountLeaves counts the number of leaves in a binary tree. For this purpose it maintains a stack S that contains all subtrees of the input tree *root* that still need to be traversed. In each iteration the first element y is removed from S. If y is a leaf then the count is increased. Otherwise the subtrees of y are pushed on the stack. Then the computation continues with the updated stack.

In order to prove termination of program CountLeaves we need to find a well-founded ordering on the states of the program that decreases with every iteration of the loop. This well-founded ordering needs to capture the fact that in each loop iteration either some tree is removed from the stack, or some tree on the stack is replaced by finitely many smaller trees. This can be naturally expressed in terms of a multiset ordering. We therefore abstract the program CountLeaves by a program over multisets.

```
prog CountLeaves(root : Tree) : int =
  var S : Stack[Tree] = root
  var c : int = 0
  do
    y := head(S)
    if leaf(y) then
      S := tail(S)
      c := c + 1
    else S := left(y) · right(y) · tail(S)
  until S = ϵ
  return c
```

Fig. 1. Program COUNTLEAVES: counting the leaves in a binary tree

```
prog AbsCountLeaves(root : Tree) =
  var X_S : multiset[Tree] = {root}
  do
    y := choose(X_S)
    if leaf(y) then X_S := X_S \ {y}
    else X_S := (X_S \ {y}) ⊎ {left(y)} ⊎ {right(y)}
  until X_S = ∅
```

Fig. 2. Multiset abstraction of program COUNTLEAVES

The result of this abstraction is shown in Figure 2. The program ABSCOUNTLEAVES is obtained from program COUNTLEAVES by mapping the stack S to a multiset X_S, i.e., in program ABSCOUNTLEAVES we abstract from the order of the elements in S. In program ABSCOUNTLEAVES the stack operations are replaced by operations on multisets. For instance, the operation $head(S)$ is abstracted by the operation $choose(X_S)$ that non-deterministically chooses an element from the multiset X_S. The computation of such multiset abstractions of programs could be automated by combining techniques developed in [27] and [5, 24]. In this paper we focus on automating the termination proofs for the resulting multiset program.

We prove termination of program ABSCOUNTLEAVES by proving that for every iteration of the loop, the variable X_S decreases in the ordering $\prec_{\mathsf{m}}$. The ordering $\prec_{\mathsf{m}}$ is the multiset extension of the subtree ordering $\prec$ on the trees stored in the multiset. The subtree ordering is well-founded; consequently, so is its multiset extension. Termination of program ABSCOUNTLEAVES is therefore implied by the validity of the *termination condition* given in Figure 3. The decision procedure presented in this paper decides the validity of such termination conditions (respectively, unsatisfiability of their negation). In Section 3 we show how the decision procedure works on a formula similar to the one shown in Figure 3.

Example 2: computing negation normal form. Our second example is a rewrite system that computes the negation normal form of a propositional formula. It consists of the three rewrite rules shown in Figure 4. The three rules are applied non-deterministically to any matching subformula.

In order to prove termination of this rewrite system, Dershowitz [8] suggested the following mapping from a propositional formula F to a multiset of natural numbers

$$X_S \neq \emptyset \wedge X_S(y) > 0 \wedge$$
$$(X'_S = X_S \setminus \{y\} \vee X'_S = (X_S \setminus \{y\}) \uplus \{\mathit{left}(y)\} \uplus \{\mathit{right}(y)\}) \rightarrow X'_S \prec_{\mathsf{m}} X_S$$

Fig. 3. Termination condition for program AbsCountLeaves

$$\begin{aligned} \neg\neg F &\rightsquigarrow F \\ \neg(F \wedge G) &\rightsquigarrow \neg F \vee \neg G \\ \neg(F \vee G) &\rightsquigarrow \neg F \wedge \neg G \end{aligned}$$

Fig. 4. Rewrite system for computing negation normal form

X_F. Let $[G]$ denote the number of operators other than $\neg$ that occur in G, then define

$$X_F = \{\, [G] \mid \neg G \text{ is a subformula of } F \,\}$$

We can then prove that, for each rewrite rule applied to a formula F, X_F decreases in the multiset extension $<_{\mathsf{m}}$ of the ordering $<$ on natural numbers. This amounts to checking validity of the following two implications:

$$X_F = X'_F \uplus \{x, x\} \rightarrow X'_F <_{\mathsf{m}} X_F$$
$$X_F = Y \uplus \{x + y + 1\} \wedge x > 0 \wedge X'_F = Y \uplus \{x, y\} \rightarrow X'_F <_{\mathsf{m}} X_F$$

Again, these checks can be automated using our decision procedure.

3 Decision Procedure through an Example

We now explain our decision procedure through an example. The decision procedure is parameterized by the theory of the base elements comprising the multisets. For instance, in the first example given in Section 2, the base theory is the theory of trees with the subtree ordering. This theory is decidable in NP [28]. In general, the base theory can be any decidable theory equipped with a preorder. Our decision procedure reduces the formula with ordering constraints over multisets to a formula containing ordering constraints on the base elements. Satisfiability of the reduced formula is then checked using the decision procedure of the base theory.

To demonstrate how our decision procedure works, we apply it to the following formula, which is a slightly generalized version of the negated termination condition given in Figure 3:

$$Y \subseteq X \;\wedge\; X' = (X \setminus Y) \uplus Z \;\wedge\; Z \prec_{\mathsf{m}} Y \;\wedge\; \neg(X' \prec_{\mathsf{m}} X) \tag{1}$$

This formula is unsatisfiable in the theory of preordered multisets (where the base theory is the theory of all preordered sets). The reduction of the formula works as follows. First we purify and flatten the input formula by introducing fresh variables for multisets and base elements to separate the multiset constraints from constraints in the base theory. In our example there are no base theory constraints. So, purification and flattening of formula (1) results in the formula:

$$Y \subseteq X \;\wedge\; X' = X_1 \uplus Z \;\wedge\; X_1 = X \setminus Y \;\wedge\; Z \prec_{\mathsf{m}} Y \;\wedge\; \neg(X' \prec_{\mathsf{m}} X)$$

The next step is to replace all multiset atoms by their point-wise definitions on the base elements. This gives the following formula:

$$
\begin{array}{l}
(\forall x.\, Y(x) \leq X(x)) \wedge \\
(\forall x.\, X'(x) = X_1(x) + Z(x)) \wedge \\
(\forall x.\, X_1(x) = \max\{X(x) - Y(x), 0\}) \wedge \\
(\exists y. Z(y) \neq Y(y)) \wedge (\forall z.\, Y(z) < Z(z) \;\rightarrow\; \exists y.\, Z(y) < Y(y) \wedge z \prec y) \wedge \\
(\,(\forall x. X'(x) = X(x)) \vee \exists x'.\, X(x') < X'(x') \;\wedge\; \forall x.\, X'(x) < X(x) \;\rightarrow\; \neg(x' \prec x)))
\end{array}
$$

Next, we skolemize all existentially quantified variables. In our example this introduces two Skolem constants e_1, e_2 and one Skolem function w. The resulting formula is:

$$
\begin{array}{l}
(\forall x.\, Y(x) \leq X(x)) \wedge \\
(\forall x.\, X'(x) = X_1(x) + Z(x)) \wedge \\
(\forall x.\, X_1(x) = \max\{X(x) - Y(x), 0\}) \wedge \\
Z(e_1) \neq Y(e_1) \wedge (\forall y'.\, Y(y') < Z(y') \;\rightarrow\; Z(w(y')) < Y(w(y')) \wedge y' \prec w(y'))) \wedge \\
(\,(\forall x. X'(x) = X(x)) \vee X(e_2) < X'(e_2) \;\wedge\; \forall x.\, X'(x) < X(x) \;\rightarrow\; \neg(e_2 \prec x))
\end{array}
$$

The idea is now to replace each remaining universal quantifier with a finite conjunction by instantiating each quantifier with finitely many ground terms generated from the constants appearing in the formula and the introduced Skolem functions. The problem is that finite instantiation is in general incomplete because the Skolem functions coming from the ordering constraints generate an infinite Herbrand universe. Before instantiation we therefore first conjoin the skolemized formula with additional axioms that further constrain the Skolem functions. In our example, we add the two axioms:

$$\forall x\, y.\, Y(x) < Z(x) \;\wedge\; w(x) \prec y \;\rightarrow\; Y(y) \leq Z(y), \quad \forall x.\, Z(x) = Y(x) \;\rightarrow\; w(x) = x$$

We will show in Section 6 that this step is sound and ensures that instantiation of the strengthened formula with the terms $e_1, e_2, w(e_1)$ and $w(e_2)$ is sufficient for proving unsatisfiability of the original constraint. The instantiated formula is a quantifier-free formula over symbols of the base theory (such as the preorder $\prec$), the theory of linear arithmetic, and the theory of free function symbols (the multisets and the Skolem functions). The satisfiability of such formulas can be decided using a Nelson-Oppen combination of the decision procedures for the corresponding component theories. In our example, the instantiated formula implies the following disjunction:

$$
\begin{array}{l}
Z(e_1) \neq Y(e_1) \wedge X'(e_1) = X(e_1) - Y(e_1) + Z(e_1) \wedge X'(e_1) = X(e_1) \vee \\
X'(e_2) = X(e_2) - Y(e_2) + Z(e_2) \wedge X(e_2) < X'(e_2) \wedge Y(e_2) \geq Z(e_2) \vee \\
X'(w(e_2)) = X(w(e_2)) - Y(w(e_2)) + Z(w(e_2)) \wedge \\
\qquad Z(w(e_2)) < Y(w(e_2)) \wedge X'(w(e_2)) \geq X(w(e_2)) \vee \\
e_2 \prec w(e_2) \wedge \neg(e_2 \prec w(e_2))
\end{array}
$$

Observe that each of the disjuncts is unsatisfiable and, hence, so is the original formula (1).

4 Preliminaries

Before we describe the logic and decision procedure for multiset orderings, we briefly fix our formal framework.

Sorted logic. A *signature* Σ is a tuple (S, Ω), where S is a countable set of sorts and Ω is a countable set of function symbols f. Every $f \in \Omega$ is associated with an arity $n \geq 0$ and a sort $s_1 \times \cdots \times s_n \to s_0$ with $s_i \in S$ for all $i \leq n$. Function symbols of arity 0 are called *constant symbols*. For the description of our problem we will consider three sorts: $S = \{\mathsf{int}, \mathsf{bool}, \mathsf{elem}\}$. We treat predicates of sort $s_1 \times \cdots \times s_n$ as function symbols of sort $s_1 \times \ldots \times s_n \to \mathsf{bool}$. We say that a signature Σ_1 extends a signature Σ_2 if Σ_1 contains at least the sorts and function symbols of Σ_2. Let V be a countably infinite set of sorted variables, disjoint from Ω. Terms are built as usual from the function symbols in Ω and variables taken from V. We denote by $t : s$ that term t has sort s. A term t is *ground*, if no variable appears in t. We denote by $\mathsf{Terms}(\Sigma)$ the set of all ground Σ-terms. An *atom* is either constructed from the equality symbol $t_1 = t_2$ applied to terms t_1 and t_2 of the same sort, or by applying a predicate symbol to terms of the respective sorts. Formulas are built from atoms as usual, using boolean connectives and quantifiers. A formula F is called *closed* or a *sentence* if no variable appears free in F.

Structures. Given a signature $\Sigma = (S, \Omega)$, a *Σ-structure* α is a function that maps each sort $s \in S$ to a non-empty set $\alpha(s)$ and each function symbol $f \in \Omega$ of sort $s_1 \times \cdots \times s_n \to s_0$ to a function $\alpha(f) : \alpha(s_1) \times \cdots \times \alpha(s_n) \to \alpha(s_0)$. Set $\alpha(s)$ is also called α-domain of the sort s. We assume that all structures interpret the sort bool by the set of Booleans $\{\mathsf{true}, \mathsf{false}\}$, and the sort int by the set of all integers $\mathbb{Z}$. The sort elem will serve as our base set for defining multisets. We speak of $\alpha(\mathsf{elem})$ simply as the *domain* of α and often identify the two.

For a Σ-structure α and a *variable assignment* $\beta : V \to \alpha(S)$, the evaluation of a term (respectively a formula) in α, β is defined as usual. In particular, we use the standard interpretations for the equality symbol and propositional connectives. A quantified variable of sort s ranges over all elements of $\alpha(s)$. For ground terms t we skip the variable assignment and simply write $\alpha(t)$ for its evaluation in α. The notions of satisfiability, validity, and entailment of formulas are also defined as usual. We write $\alpha, \beta \models F$ if α satisfies F under β. Similarly, we write $\alpha \models F$ if α satisfies F for all variable assignments β. In this case we also call α a *model* of F.

Theories. A *Σ-theory* $\mathcal{T}$ for a signature Σ is simply a set of Σ-structures. Sometimes we identify a theory by a set of Σ-sentences $\mathcal{K}$, meaning the set of all Σ-models of $\mathcal{K}$. We then call $\mathcal{K}$ the *axioms* of the theory. The satisfiability problem for a Σ-theory $\mathcal{T}$ and a set of Σ-formulas $\mathcal{F}$ is to decide whether a given $F \in \mathcal{F}$ is satisfiable in some structure of $\mathcal{T}$. If the set of formulas $\mathcal{F}$ is clear from the context, we simply speak of the satisfiability problem of the theory $\mathcal{T}$. A Σ_2-theory $\mathcal{T}_2$ is an *extension* of a Σ_1-theory $\mathcal{T}_1$ if Σ_2 is an extension of Σ_1 and for every $\alpha \in \mathcal{T}_2$, the restriction $\alpha|_{\Sigma_1}$ of α to the sorts and symbols of Σ_1 is a structure in $\mathcal{T}_1$. A Σ-theory $\mathcal{T}$ is called *stably infinite* with respect to a set of formulas $\mathcal{F}$, if for every formula $F \in \mathcal{F}$ which is satisfiable in $\mathcal{T}$, there exists a model α of F in $\mathcal{T}$, such that the domain of α has infinite cardinality.

5 POSSUM: Multiset Constraints over Preordered Sets

In this section we formally define the constraints whose satisfiability we study in this paper. Before we define the syntax and semantics of these constraints, we first define the theories of preordered sets and their extensions to finite preordered multisets.

5.1 Finite Multisets over Preordered Sets

We assume that Σ_{elem} is a signature containing at least the binary predicate symbol $\preceq$ over sort elem. Let $\mathcal{F}_{\mathsf{elem}}$ be the set of all quantifier-free ground formulas over signature Σ_{elem}. We will use the formula $t_1 \prec t_2$ as syntactic shorthand for the formula $t_1 \neq t_2 \wedge t_1 \preceq t_2$. A binary relation R defined on a set E, such that R is reflexive and transitive is called a *preorder* and set (E, R) is called a preordered set. A theory of preordered sets $\mathcal{T}_{\mathsf{elem}}$ is a Σ_{elem}-theory such that for all structures $\alpha \in \mathcal{T}_{\mathsf{elem}}$, $(\alpha(\mathsf{elem}), \alpha(\preceq))$ is a preordered set, i.e., every structure $\alpha \in \mathcal{T}_{\mathsf{elem}}$ satisfies the following two axioms:

$$\forall x : \mathsf{elem}.\ x \preceq x \quad (\mathsf{refl}) \qquad \forall x, y, z : \mathsf{elem}.\ x \preceq y \wedge y \preceq z \rightarrow x \preceq z \quad (\mathsf{trans})$$

For the rest of this paper, we fix such a theory $\mathcal{T}_{\mathsf{elem}}$. We require that the satisfiability problem for $\mathcal{F}_{\mathsf{elem}}$ and $\mathcal{T}_{\mathsf{elem}}$ is decidable. We further require that $\mathcal{T}_{\mathsf{elem}}$ is stably-infinite with respect to the formulas $\mathcal{F}_{\mathsf{elem}}$. We call $\mathcal{T}_{\mathsf{elem}}$ the *base theory*.

Let Ω_{la} be the function and constant symbols of linear integer arithmetic

$$\Omega_{\mathsf{la}} = \{+, -, \max, \min, \ldots, -2, -1, 0, 1, 2, \ldots, -2\cdot, -1\cdot, 0\cdot, 1\cdot, 2\cdot\}$$

with their appropriate sorts (the function symbol $C\cdot$ denotes multiplication with integer constant C). We assume that these symbols are disjoint from the symbols in Σ_{elem}. We represent multisets as function symbols of sort $\mathsf{elem} \rightarrow \mathsf{int}$. Let $\mathcal{M}$ be a countably infinite set of function symbols of this sort, disjoint from the symbols in Σ_{elem} and Ω_{la}. Further, let Σ_{mset} be the signature Σ_{elem} extended with the symbols $\mathcal{M}$ and Ω_{la}. We then define the theory $\mathcal{T}_{\mathsf{mset}}$ of finite preordered multisets over $\mathcal{T}_{\mathsf{elem}}$ as follows. The theory $\mathcal{T}_{\mathsf{mset}}$ is the set of all structures α such that α is an extension of a structure in $\mathcal{T}_{\mathsf{elem}}$ to a Σ_{mset}-structure and α satisfies the following conditions:

- α gives the standard interpretation to the arithmetic symbols, and
- α interprets each $X \in \mathcal{M}$ as a finite multiset, i.e., (1) for all $e \in \alpha(\mathsf{elem})$, $\alpha(X)(e) \geq 0$ and (2) there are only finitely many $e \in \alpha(\mathsf{elem})$ such that $\alpha(X)(e) > 0$.

5.2 Syntax and Semantics of POSSUM Formulas

Syntax. Figure 5 defines the POSSUM formulas. A POSSUM formula is an arbitrary propositional combination of atomic formulas. The atomic formulas are relations between multiset expressions, relations between arithmetic expressions, atoms over the base signature Σ_{elem}, and restricted quantified formulas $\mathsf{F}^{\forall}$. An example of a base signature atom is the formula $e_1 \preceq e_2$, where e_1 and e_2 are two constants of sort elem. The formulas $\mathsf{F}^{\forall}$ express universal quantification over variables of sort elem. The formulas below the quantifiers can express arithmetic relations between multiplicities $X(x)$ of the quantified variables or ordering constraints between these variables. Using these quantified formulas we can, for instance, express that some constant e is maximal in a multiset X: $\forall x.\, X(x) > 0 \rightarrow x \preceq e$. The important restriction for the formulas below the universal quantifiers is that the quantified variables x are not allowed to appear below function symbols of the base signature Σ_{elem}. This is enforced by allowing only

top-level formulas:

$$
\begin{aligned}
F &::= A \mid F \wedge F \mid \neg F \\
A &::= M = M \mid M \subseteq M \mid K = K \mid K \leq K \mid M \preceq_{\mathsf{m}} M \mid A_{\mathsf{elem}} \mid \mathsf{F}^{\forall} \\
M &::= X \mid \emptyset \mid \{t^K\} \mid M \cap M \mid M \cup M \mid M \uplus M \mid M \setminus M \mid \mathsf{setOf}(M) \\
K &::= k \mid C \mid K + K \mid C \cdot K
\end{aligned}
$$

restricted quantified formulas:

$$
\begin{aligned}
\mathsf{F}^{\forall} &::= \forall x : \mathsf{elem}.\mathsf{F}^{\forall} \mid \forall x : \mathsf{elem}.\mathsf{F}^{\mathsf{in}} \\
\mathsf{F}^{\mathsf{in}} &::= \mathsf{A}^{\mathsf{in}} \mid \mathsf{F}^{\mathsf{in}} \wedge \mathsf{F}^{\mathsf{in}} \mid \neg \mathsf{F}^{\mathsf{in}} \\
\mathsf{A}^{\mathsf{in}} &::= \mathsf{t}^{\mathsf{in}} \leq \mathsf{t}^{\mathsf{in}} \mid \mathsf{t}^{\mathsf{in}} = \mathsf{t}^{\mathsf{in}} \mid \mathsf{E}^{\mathsf{in}} \preceq \mathsf{E}^{\mathsf{in}} \mid \mathsf{E}^{\mathsf{in}} = \mathsf{E}^{\mathsf{in}} \\
\mathsf{t}^{\mathsf{in}} &::= X(\mathsf{E}^{\mathsf{in}}) \mid C \mid \mathsf{t}^{\mathsf{in}} + \mathsf{t}^{\mathsf{in}} \mid C \cdot \mathsf{t}^{\mathsf{in}} \\
\mathsf{E}^{\mathsf{in}} &::= x \mid t
\end{aligned}
$$

terminals:

A_{elem} - ground Σ_{elem}-atom ; X - multiset ; k - integer variable; C - integer constant
t - ground Σ_{elem}-term of sort elem; x - variable of sort elem

Fig. 5. Syntax for Multiset Constraints over Preordered Sets (POSSUM)

ground Σ_{elem}-terms t below the quantifiers. Note also that there are no POSSUM formulas with $\mathsf{F}^{\forall}$ atoms that have an alternating quantifier prefix. We call a subset $\mathcal{F}$ of POSSUM formulas *quantifier-bounded* if the number of quantified variables appearing in $\mathsf{F}^{\forall}$ subformulas of formulas in $\mathcal{F}$ is bounded.

Semantics. POSSUM formulas are interpreted in the structures of the theory $\mathcal{T}_{\mathsf{mset}}$. The semantics of POSSUM formulas extends the semantics of first-order formulas defined in Section 4. Note that with the exception of atomic formulas that express relations on multisets, all atomic formulas are first-order formulas. Thus, we only need to define the semantics of formulas of the form $M_1 = M_2$, $M_1 \subseteq M_2$, and $M_1 \preceq_{\mathsf{m}} M_2$. Let α be a structure in $\mathcal{T}_{\mathsf{mset}}$. First, we extend the interpretation $\alpha(X)$ of multisets $X \in \mathcal{M}$ in α to multiset expressions. The interpretation is defined point-wise for all $e \in \alpha$ and recursively on the structure of the expression:

$$
\begin{aligned}
\alpha(\emptyset)(e) &= 0 \\
\alpha(\{t^K\})(e) &= \text{if } \alpha(t) = e \text{ then } \alpha(K) \text{ else } 0 \\
\alpha(M_1 \cup M_2)(e) &= \max\{\alpha(M_1)(e), \alpha(M_2)(e)\} \\
\alpha(M_1 \cap M_2)(e) &= \min\{\alpha(M_1)(e), \alpha(M_2)(e)\} \\
\alpha(M_1 \uplus M_2)(e) &= \alpha(M_1)(e) + \alpha(M_2)(e) \\
\alpha(M_1 \setminus M_2)(e) &= \max\{\alpha(M_1)(e) - \alpha(M_2)(e), 0\} \\
\alpha(\mathsf{setOf}(M))(e) &= \min\{\alpha(M)(e), 1\}
\end{aligned}
$$

For defining the interpretations of the predicate symbols $=$, $\subseteq$, and $\preceq_{\mathsf{m}}$ on multisets we define corresponding relations $=_{\mathsf{m}}$, $\subseteq_{\mathsf{m}}$, and $\preceq_{\mathsf{m}}$ at the meta-level. Let m_1, m_2 be functions $\alpha(\mathsf{elem}) \to \mathbb{N}$. The relations $=_{\mathsf{m}}$ and $\subseteq_{\mathsf{m}}$ are defined point-wise as expected:

$$
\begin{aligned}
m_1 =_{\mathsf{m}} m_2 \quad &\Leftrightarrow \quad \forall e \in \alpha(\mathsf{elem}).m_1(e) = m_2(e) \\
m_1 \subseteq_{\mathsf{m}} m_2 \quad &\Leftrightarrow \quad \forall e \in \alpha(\mathsf{elem}).m_1(e) \leq m_2(e)
\end{aligned}
$$

For defining the multiset ordering we identify $\prec$ with the irreflexive reduct of the relation $\alpha(\preceq)$. The relation $\preceq_{\mathsf{m}}$ is then defined as follows:

$$m_1 \preceq_{\mathsf{m}} m_2 \quad \Leftrightarrow \quad \forall e_1 \in \alpha.\, m_1(e_1) > m_2(e_1) \Rightarrow \exists e_2 \in \alpha. m_2(e_2) > m_1(e_2) \wedge e_1 \prec e_2 \tag{2}$$

Note that this is not the standard definition of the multiset ordering that was originally used in [9]. However, in order to reduce the number of multiset variables we use the simpler definition (2). For finite multisets, definition (2) is equivalent to the standard one (for proof see [1, Lemma 2.5.6, p.24]).

6 Decision Procedure

We now describe the decision procedure for POSSUM. The idea of the decision procedure is to reduce satisfiability of a POSSUM formula to satisfiability of a formula in a particular first-order theory, namely, the disjoint combination of the base theory $\mathcal{T}_{\mathsf{elem}}$, the theory of linear integer arithmetic, and the theory of uninterpreted function symbols.

Reduction to a first-order theory. In the following, we show how to decide conjunctions of POSSUM literals. The extension of the decision procedure to arbitrary Boolean combinations of literals is straightforward. Thus, let F be a fixed POSSUM conjunction. The first step of our decision procedure is to rewrite F into a quantified first-order formula by expanding all multiset constraints to their point-wise definitions.

For two multiset variables X and Y we denote by $L_{X,Y}$ the multiset $X \setminus Y$ and by $U_{X,Y}$ the multiset $Y \setminus X$. Similarly, for a given element x we use $L_{X,Y}(x)$ as a shorthand for the expression $X(x) - Y(x)$ and $U_{X,Y}(x)$ for $Y(x) - X(x)$. The algorithm for rewriting F is then as follows:

1. Purify and flatten all multiset constraints in F:
   $C[M] \rightsquigarrow X_f = M \wedge C[X_f]$
   where $X_f \in \mathcal{M}$ is a fresh multiset and M is
   (a) either of the form $M_1 \cup M_2$, $M_1 \cap M_2$, $M_1 \uplus M_2$, $M_1 \setminus M_2$, and at least one M_i is not a multiset $X \in \mathcal{M}$
   (b) or of the form $\emptyset$, $\{t^k\}$, $\mathsf{setOf}(M_1)$, and they are not in a conjunct of the form $X = M$ or $M = X$ for some multiset $X \in \mathcal{M}$.
2. Replace all multiset atoms by their point-wise definitions

$$\begin{array}{ll}
C[X = \emptyset] & \rightsquigarrow\ C[\forall x.\, X(x) = 0] \\
C[X = \{e^k\}] & \rightsquigarrow\ C[X(e) = k \wedge \forall x.\, x \neq e \rightarrow X(e) = 0] \\
C[X = Y \cup Z] & \rightsquigarrow\ C[\forall x.\, X(x) = \max\{Y(x), Z(x)\}] \\
C[X = Y \cap Z] & \rightsquigarrow\ C[\forall x.\, X(x) = \min\{Y(x), Z(x)\}] \\
C[X = Y \uplus Z] & \rightsquigarrow\ C[\forall x.\, X(x) = Y(x) + Z(x)] \\
C[X = Y \setminus Z] & \rightsquigarrow\ C[\forall x.\, X(x) = \max\{Y(x) - Z(x), 0\}] \\
C[X = Y] & \rightsquigarrow\ C[\forall x.\, X(x) = Y(x)] \\
C[X \preceq_{\mathsf{m}} Y] & \rightsquigarrow\ C[\forall x.\, L_{X,Y}(x) > 0 \rightarrow \exists y.\, U_{X,Y}(y) > 0 \wedge x \prec y]
\end{array}$$

3. Compute negation normal form, i.e., push all negations down to the atoms

4. Skolemize all existentially quantified variables
5. For every multiset X occurring in the formula add the formula $\forall x.\, X(x) \geq 0$ as an additional conjunct

After rewriting, the resulting formula is of the form $\mathcal{K} \wedge G$ where G is a ground formula and $\mathcal{K}$ is a conjunction of universally quantified formulas. Clearly, each step of the rewriting transforms the input formula into an equisatisfiable formula.

Lemma 1. *The formulas F and $\mathcal{K} \wedge G$ are equisatisfiable in the theory of preordered multisets.*

Quantifier instantiation. We will now show that there exists a finite and computable set of ground terms $T_{\mathcal{K},G}$ of sort elem such that $\mathcal{K} \wedge G$ is equisatisfiable to the formula $\mathcal{K}[T_{\mathcal{K},G}] \wedge G$, where $\mathcal{K}[T_{\mathcal{K},G}]$ is a ground formula obtained by instantiating all quantified variables appearing in $\mathcal{K}$ with the terms in $T_{\mathcal{K},G}$.

Throughout the rest of this section we denote by E the set of all ground terms of sort elem appearing in $\mathcal{K} \wedge G$. The set E contains the ground terms appearing in the initial formula F and Skolem constants that have been introduced for top-level existentially quantified variables in Step 4 of the rewrite algorithm. Zarba showed in [29] that for formulas F without ordering constraints on multisets and formulas $\mathsf{F}^{\forall}$, the theory $\mathcal{K}$ is (what is now known as) a *stably local theory extension* [25]. This means that if F does not contain ordering constraints then F is equisatisfiable to the formula $\mathcal{K}[E] \wedge G$. The reason for locality of $\mathcal{K}$ in this case is simply that instantiation of the quantifiers in $\mathcal{K}$ with terms of sort elem will not create new terms of the same sort. Unfortunately, in the presence of ordering constraints this is no longer true, i.e., instantiation of $\mathcal{K}$ with the terms in E alone is not sufficient.

For illustration of this behavior, reconsider the defining formula (2) for the ordering constraint $X \preceq_{\mathsf{m}} Y$. This formula contains $\forall\exists$ quantification over variables of sort elem. Skolemization of this formula thus gives

$$\forall x.\, L_{X,Y}(x) > 0 \;\rightarrow\; U_{X,Y}(w_{X,Y}(x)) > 0 \wedge x \prec w_{X,Y}(x) \tag{3}$$

where $w_{X,Y}$ is a fresh Skolem function. We call these Skolem functions $\prec_{\mathsf{m}}$-*witness functions* and terms constructed from these functions $\prec_{\mathsf{m}}$-*witnesses*. Instantiation of formula (3) with a term $e \in E$ generates a new $\prec_{\mathsf{m}}$-witness $w_{X,Y}(e)$ of sort elem, which is not already contained in E. For completeness we have to instantiate $\mathcal{K}$ recursively with these $\prec_{\mathsf{m}}$-witnesses.

We now show that we can put additional constraints on the $\prec_{\mathsf{m}}$-witness functions such that we only need to consider finitely many $\prec_{\mathsf{m}}$-witnesses for the instantiation of $\mathcal{K}$. These additional constraints are as follows. First, we enforce that the $\prec_{\mathsf{m}}$-witness function $w_{X,Y}$ only chooses maximal elements in the multiset $U_{X,Y}$ and, second, we require that each element outside $L_{X,Y}$ is mapped to itself. Formally, these constraints are expressed by the following two axioms:

$$\forall x\, y.\, L_{X,Y}(x) > 0 \wedge w_{X,Y}(x) \prec y \;\rightarrow\; U_{X,Y}(y) = 0 \tag{4}$$

$$\forall x.\, L_{X,Y}(x) = 0 \;\rightarrow\; w_{X,Y}(x) = x \tag{5}$$

The existence of such constrained witness functions is guaranteed by the fact that we restrict ourselves to finite multisets. In particular, given a $\prec_{\mathsf{m}}$-witness function $w_{X,Y}$ satisfying axiom (3), we can define a new witness function that maps every e in $L_{X,Y}$ to the maximal element of some ascending chain starting from $w_{X,Y}(e)$ in $U_{X,Y}$. Finiteness of the multiset $U_{X,Y}$ guarantees the existence of such a maximal element.

For the rest of this section let W be the set of all $\prec_{\mathsf{m}}$-witness functions occurring in $\mathcal{K}$ and let $\mathcal{K}_W$ be the conjunction of axioms (4) and (5) for all $w_{X,Y} \in W$.

Lemma 2. *The formulas $\mathcal{K} \wedge G$ and $\mathcal{K} \wedge \mathcal{K}_W \wedge G$ are equisatisfiable in the theory of preordered multisets.* □

Let $T_{W,E}$ be the smallest set of ground terms that satisfies the following two conditions: (i) $E \subseteq T_{W,E}$ and (ii) if $t \in T_{W,E}$ and $w_{X,Y} \in W$ then $w_{X,Y}(t) \in T_{W,E}$. For a term $t \in T_{W,E}$ of the form $t = w_n \dots w_1(e)$ where $e \in S$, we define $t_0 = e$ and denote by t_i, for $1 \leq i \leq n$, the subterm $w_i \dots w_1(e)$ of t. We call $t \in T_{W,E}$ a strict chain in a structure α iff α satisfies $t_i \prec t_{i+1}$ for all i with $0 \leq i < n$. We say that a strict chain $t \in T_{W,E}$ in a structure α is *maximal* if t is not a proper subterm of any other strict chain $t' \in T_{W,E}$ in α. For a structure α and a set of ground terms T, we denote by $\alpha(T)$ the set $\alpha(T) = \{\alpha(t) \mid t \in T\}$.

Now, define $T_{\mathcal{K},G}$ as the set of all terms $t \in T_{W,E}$ such that each function $w_{X,Y}$ occurs at most once in t. Clearly, the set $T_{\mathcal{K},G}$ is finite, since W is finite. We can now show that in models of $\mathcal{K} \wedge \mathcal{K}_W$, the terms $T_{W,E}$ are partitioned into finitely many equivalence classes, each of which is represented by some term in $T_{\mathcal{K},G}$.

Lemma 3. *For all models α of $\mathcal{K} \wedge \mathcal{K}_W$, $\alpha(T_{W,E}) = \alpha(T_{\mathcal{K},G})$.*

Proof. Let α be a model of $\mathcal{K} \wedge \mathcal{K}_W$. Note that from strictness of $\prec$, and axioms (3) and (5) it follows that for all terms t of sort elem and $w_{X,Y} \in W$, either $\alpha \models w_{X,Y}(t) = t$ or $\alpha \models t \prec w_{X,Y}(t)$ holds.

The proof goes by contradiction. Thus, assume there exists $t \in T_{W,E}$ such that $\alpha(t) \notin \alpha(T_{\mathcal{K},G})$. Then remove all function applications w_i from t for which $\alpha \models w_i(t_{i-1}) = t_{i-1}$, obtaining a term $t' \in T_{W,E}$. Then t' is a strict chain and $\alpha \models t = t'$. From this we conclude that $\alpha(t') \notin \alpha(T_{\mathcal{K},G})$ and therefore $t' \notin T_{\mathcal{K},G}$. Hence, there exists i, j with $1 \leq i < j < k$ and multiset variables X, Y such that $w'_i = w'_j = w_{X,Y} \in W$. We then have $\alpha \models t'_{i-1} \prec w_{X,Y}(t'_{i-1})$. Based on strictness of $\prec$, axiom (5) and axiom $\forall x. L_{X,Y}(x) \geq 0$ we conclude $\alpha \models L_{X,Y}(t'_{i-1}) > 0$. Similarly, we conclude $\alpha \models L_{X,Y}(t'_{j-1}) > 0$. By transitivity of $\prec$ and construction of t', we further have that α satisfies $w_{X,Y}(t'_{i-1}) \prec w_{X,Y}(t'_{j-1})$. From axiom (4) we then conclude $\alpha \models U_{X,Y}(w_{X,Y}(t'_{j-1})) = 0$. However, axiom (3) implies $\alpha \models U_{X,Y}(w_{X,Y}(t'_{j-1})) > 0$, which gives us a contradiction. □

From Lemma 3 it follows that we only need to instantiate the axioms $\mathcal{K} \wedge \mathcal{K}_W$ with the terms in $T_{\mathcal{K},G}$.

Lemma 4. *The formulas $\mathcal{K} \wedge \mathcal{K}_W \wedge G$ and $\mathcal{K}[T_{\mathcal{K},G}] \wedge \mathcal{K}_W[T_{\mathcal{K},G}] \wedge G$ are equisatisfiable in the theory of preordered multisets.*

The formula $\mathcal{K}[T_{\mathcal{K},G}] \wedge \mathcal{K}_W[T_{\mathcal{K},G}] \wedge G$ can now be purified obtaining an equisatisfiable formula $G_{\mathsf{elem}} \wedge G_{\mathsf{la}} \wedge G_{\mathsf{euf}}$ such that the three conjuncts G_{elem}, G_{la}, and G_{euf} only share constant symbols and:

- G_{elem} is a constraint over symbols in the theory $\mathcal{T}_{\mathsf{elem}}$
- G_{la} is a linear integer arithmetic constraint, and
- G_{euf} is a constraint built from uninterpreted function symbols and equality

We can thus check satisfiability of F by checking satisfiability of $G_{\mathsf{elem}} \wedge G_{\mathsf{la}} \wedge G_{\mathsf{euf}}$ in the disjoint combination of the theory $\mathcal{T}_{\mathsf{elem}}$, the theory of linear integer arithmetic, and the theory of uninterpreted function symbols with equality. By our assumptions on the theory $\mathcal{T}_{\mathsf{elem}}$, this combined theory can be decided using standard Nelson-Oppen combination techniques [18].

Theorem 1. *The satisfiability problem for* POSSUM *formulas is decidable.*

Complexity. We will now establish that the satisfiability problem for the quantifier-bounded fragments of POSSUM is in NP, provided the base theory $\mathcal{T}_{\mathsf{elem}}$ is also decidable in NP. Since POSSUM formulas subsume propositional logic this bound is tight.

We have seen in the previous section that we can reduce a POSSUM conjunction F to a ground formula $\mathcal{K}[T_{\mathcal{K},G}] \wedge \mathcal{K}_W[T_{\mathcal{K},G}] \wedge G$ whose satisfiability can be decided using the decision procedure of the base theory. However, the size of the resulting formula can be exponential in the size of the input formula F because the size of the set $T_{\mathcal{K},G}$ used for the instantiation is exponential in the number of $\prec_{\mathsf{m}}$-witness functions W. The following lemma implies that this exponential blowup can be avoided.

Lemma 5. *If the formula $\mathcal{K} \wedge \mathcal{K}_W \wedge G$ is satisfiable then it has a model α such that $|\alpha(T_{\mathcal{K},G})| \in \mathcal{O}(|W|^2 \cdot |E|)$.*

Proof. Assume $\mathcal{K} \wedge \mathcal{K}_W \wedge G$ is satisfiable and let α_0 be one of its models. Further, let $n = |W|$ and $m = |E|$. From α_0 we construct a model α with $|\alpha(T_{\mathcal{K},G})| \in \mathcal{O}(n^2 m)$ by collapsing redundant strict chains in α_0. For this purpose, we choose a set T of strict chains in α_0 such that for every term $e \in E$ and witness function $w_{X,Y} \in W$, there is at most one chain $t \in T$ that starts in $L_{X,Y}$, i.e., t contains $w_{X,Y}(e)$ as a subterm. Formally, let $E_=$ be the quotient of E with respect to the interpretation of the equality predicate $=$ in α_0 and denote by $[e] \in E_=$ the equivalence class of $e \in E$. Let T be a maximal subset T of $T_{K,G}$ such that (i) each $t \in T$ is a maximal strict chain in α_0, and (ii) for each $w \in W$, if there is some $t \in T$ which contains w and starts in $e \in E$ then there is no other $t' \in T$ which contains $w(e')$ as a subterm, for any $e' \in [e]$. Clearly such a set T exists. Let T^* be the set of all subterms $t_0, \ldots, t_k$ of the chains $t \in T$, where k is the length of chain t.

We now construct α from α_0 by collapsing all strict chains in α_0 to the chains in T. First, we let α agree with α_0 on the interpretation of all sorts and all symbols that are not witness functions. For each witness function $w \in W$ and $v \in \alpha(\mathsf{elem})$, we then define $\alpha(w)(v)$ as follows: if $v = \alpha_0(e)$ for some $e \in E$, $\alpha_0(w(e)) \neq \alpha_0(e)$, and there is some term $t \in T^*$ with $t = w(t')$ for some t' containing $e' \in [e]$, choose one such term t and define $\alpha(w)(v) = \alpha_0(t)$. In all other cases define $\alpha(w)(v) = \alpha_0(w)(v)$.

Note that from the definition of T and α it follows that for all $t \in T^*$, $\alpha(t) = \alpha_0(t)$. Thus all terms in T^* are still strict chains in α.

We first prove that α is still a model of $\mathcal{K} \wedge \mathcal{K}_W \wedge G$. Since α_0 is a model of $\mathcal{K} \wedge \mathcal{K}_W \wedge G$ and α agrees with α_0 on all symbols that are not witness functions, we immediately conclude that α is also a model of G and all axioms of $\mathcal{K}$ that do not mention the witness functions. The fact that α still satisfies the remaining axioms (3)-(5) for all $w \in W$ also easily follows from the definition of α. In particular, if for some w and v, $\alpha(w)(v) \neq \alpha_0(w)(v)$ then, by definition, $v = \alpha(e)$ for some $e \in E$ and $\alpha(w)(v) = \alpha(t)$ for some term $t \in T^*$ such that t contains $e' \in [e]$. The fact that t is a strict chain in α starting in e' and the transitivity of $\prec$ imply $\alpha \models e' \prec t$. Since $e' \in [e]$ we further have $\alpha \models e = e'$ and hence $\alpha \models e \prec w(e)$. Thus α still satisfies axiom (3). The proofs for the other two axioms are similar.

For proving that $|\alpha(T_{\mathcal{K},G})| \in \mathcal{O}(n^2m)$ first note that by construction of α we have $\alpha(T^*) = \alpha(T_{K,G})$. We thus need to count the number of elements in T^*. For this purpose, fix $e \in E$ and let k be the maximal length of the chains t in T that start from some $e' \in [e]$. From strictness of the chains and Lemma 3 it follows that $k \leq n$. We then have by condition (ii) of the definition of T that there are at most $n-k+1$ chains t in T that start from some $e' \in [e]$. Each of these chains has at most length k by assumption and thus at most $k+1$ subterms. It follows that T^* contains at most $(n-k+1)(k+1)$ terms with some $e' \in [e]$ as a subterm. From $\max_{1 \leq k \leq n} \{(n-k+1)(k+1)\} \in \mathcal{O}(n^2)$ we then conclude $|T^*| \in \mathcal{O}(n^2m)$. □

Lemma 5 implies that we can guess a polynomial subset T of the terms $T_{K,G}$ and then use this subset to instantiate the axioms in $\mathcal{K} \wedge \mathcal{K}_W$. The size of the resulting formula $\mathcal{K}[T] \wedge \mathcal{K}_W[T] \wedge G$ is then polynomial in the size of the input formula, provided we bound the number of quantified variables in $\mathsf{F}^\forall$ subformulas of the input.

Theorem 2. *If the base theory $\mathcal{T}_{\mathsf{elem}}$ is decidable in NP then for the quantifier-bounded fragments of its* POSSUM *extension, the satisfiability problem is NP-complete.*

Practical Considerations. Our decision procedure is amenable to practical implementations using off-the-shelf SMT-solvers. In particular, using techniques developed for local theory extensions [13], we can postpone the exponential decomposition phase of guessing the terms used for instantiation, by generating these terms lazily from models produced by the SMT solver. Also note that in practical applications such as checking validity of constraints generated from termination proofs, all multiset ordering constraints $X \prec_{\mathsf{m}} Y$ will typically have negative polarity. Since only positive occurrences of such constraints generate $\prec_{\mathsf{m}}$-witness functions, the set of terms $T_{K,G}$ will, in most practical cases, already be polynomial in the size of the input constraint.

7 Related Work

The logic POSSUM extends the logic of multisets with integers, which was shown to be NP-complete by Zarba [29]. This extension is non-trivial. In particular, Zarba only considers a disjoint combination of a base theory with the theory of multisets and does not support ordering constraints on multisets. Such constraints generate axioms with $\forall\exists$

quantification, which require a more intricate argument to establish completeness of local instantiation. The logic of multisets with cardinality constraints [20] also subsumes Zarba's logic and was shown to be NP-complete [21]. It is incomparable to our logic because it also does not support ordering constraints. On the other hand, POSSUM can only express very restricted cardinality constraints. In [14] the theory of sets with cardinality constraints over totally ordered base sets was shown to be decidable in NP. This result can be generalized to multisets. Decidability of multisets over partially ordered base sets and with general cardinality constraints is open.

Local theory extensions [25] formalize the general category of theories for which local quantifier instantiation techniques are complete. Some local theory extension of orders have been studied in [26]. Our extension of preorders to multiset orderings is an instance of the so called Ψ-local theory extensions, which have been introduced in [12].

Simplification orderings are a common tool to prove termination of term rewrite systems [1, 8]. Among the most widely used simplification orderings are recursive path orderings [8] (which have originally been defined in terms of multiset orderings), lexicographic path orderings [1], and Knuth-Bendix orderings [10]. Constraint solving has been shown to be decidable in NP for each of these orderings [17, 19, 30]. Unlike simplification orderings, we do not require that the underlying order is total. Thus, one can use our decision procedure to prove termination even in cases where there are no natural total orderings, such as Example 1 in Section 2.

8 Conclusion

We presented POSSUM, a new logic and decision procedure for reasoning about multiset orderings. POSSUM can express constraints over complex well-founded orderings, which makes it a useful tool for proving termination. The logic subsumes linear integer arithmetic which has been traditionally used to express ranking functions in automated termination proofs. We established that the satisfiability problem for POSSUM is NP-complete, provided the base theory is in NP. Thus it has the same complexity as quantifier-free linear integer arithmetic. Furthermore, our decision procedure is amenable to a practical implementation. We thus believe that POSSUM provides a valuable tool for extending the scope of existing termination provers. Our next step is to implement the decision procedure and make it available as a component for SMT solvers.

Acknowledgments. We thank Viktor Kuncak for inspiring discussions and his valuable comments on an earlier draft of this paper.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. L. Bachmair and H. Ganzinger. Resolution theorem proving. In *Handbook of Automated Reasoning*, pages 19–99. MIT Press, 2001.
3. J. Berdine, B. Cook, D. Distefano, and P. W. O'Hearn. Automatic termination proofs for programs with shape-shifting heaps. In *CAV*, pages 386–400, 2006.
4. M. Colón and H. Sipma. Synthesis of linear ranking functions. In *TACAS*, pages 67–81, 2001.

5. B. Cook, A. Podelski, and A. Rybalchenko. Abstraction refinement for termination. In *SAS*, pages 87–101, 2005.
6. B. Cook, A. Podelski, and A. Rybalchenko. Terminator: Beyond safety. In *CAV*, pages 415–418, 2006.
7. Y. Deng and D. Sangiorgi. Ensuring termination by typability. *Inf. Comput.*, 204(7):1045–1082, 2006.
8. N. Dershowitz. Orderings for term-rewriting systems. In *Symposium on Foundations of Computer Science (SFCS)*, pages 123–131, 1979.
9. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Commun. ACM*, 22(8):465–476, 1979.
10. J. Dick, J. Kalmus, and U. Martin. Automating the Knuth Bendix Ordering. *Acta Inf.*, 28(2):95–119, 1990.
11. R. W. Floyd. Assigning meanings to programs. In *Proc. Amer. Math. Soc. Symposia in Applied Mathematics*, volume 19, pages 19–31, 1967.
12. C. Ihlemann, S. Jacobs, and V. Sofronie-Stokkermans. On local reasoning in verification. In *TACAS*, pages 265–281, 2008.
13. S. Jacobs. Incremental instance generation in local reasoning. In *CAV*, pages 368–382, 2009.
14. V. Kuncak, R. Piskac, and P. Suter. Ordered sets in the calculus of data structures. In *CSL*, pages 34–48, 2010.
15. C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. In *POPL*, pages 81–92, 2001.
16. F.-J. Martín-Mateos, J.-L. Ruiz-Reina, J.-A. Alonso, and M.-J. Hidalgo. Proof Pearl: A Formal Proof of Higman's Lemma in ACL2. In *TPHOLs*, pages 358–372, 2005.
17. P. Narendran, M. Rusinowitch, and R. M. Verma. RPO Constraint Solving is in NP. In *CSL*, pages 385–398, 1998.
18. G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM TOPLAS*, 1(2):245–257, 1979.
19. R. Nieuwenhuis. Simple LPO constraint solving methods. *Inf. Process. Lett.*, 47(2):65–69, 1993.
20. R. Piskac and V. Kuncak. Decision procedures for multisets with cardinality constraints. In *VMCAI*, number 4905 in LNCS, 2008.
21. R. Piskac and V. Kuncak. Linear arithmetic with stars. In *CAV*, 2008.
22. A. Podelski and A. Rybalchenko. A complete method for synthesis of linear ranking functions. In *VMCAI'04*, 2004.
23. A. Podelski and A. Rybalchenko. Transition invariants. In *LICS'04*, 2004.
24. A. Podelski and A. Rybalchenko. Transition predicate abstraction and fair termination. *ACM TOPLAS*, 29(3):15, 2007.
25. V. Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. In *CADE*, pages 219–234, 2005.
26. V. Sofronie-Stokkermans and C. Ihlemann. Automated reasoning in some local extensions of ordered structures. In *ISMVL*, 2007.
27. P. Suter, M. Dotta, and V. Kuncak. Decision procedures for algebraic data types with abstractions. In *37th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 2010.
28. K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *Journal of the ACM (JACM)*, 34(2):492–510, 1987.
29. C. G. Zarba. Combining multisets with integers. In *CADE-18*, 2002.
30. T. Zhang, H. B. Sipma, and Z. Manna. The Decidability of the First-Order Theory of Knuth-Bendix Order. In *CADE*, pages 131–148, 2005.