# Arithmetic Integration of Decision Procedures*

Ting Zhang

Stanford University

## Chapter 1. Introduction

Decision procedures are algorithms that can determine whether a formula in a given logical theory is valid or satisfiable. An important distinction between these algorithms and general-purpose theorem provers is the level of automation. Decision procedures are fully automatic; they always terminate with either positive or negative answers. In formal program verification, hundreds of thousands of verification conditions are generated. Program verification will never be practical if human efforts are needed to prove these formulas, most of which are repetitive, tediously long and mathematically uninteresting. Decision procedures can automatically discharge verification conditions that fall in the scope of certain decidable theories, and hence relieve users of verification systems from tedious interaction with general-purpose theorem provers. Besides their indispensable role in rigorous formal verification, decision procedures are also of great importance in model checking and program analysis where automatic proof engines can improve the overall efficiency and increase the analysis accuracy.

Decision procedures exist for many specialized logical domains including integers, reals as well as for many data structures frequently appearing in programs such as lists, queues, sets and multi-sets. These specialized decision procedures can only handle a particular class of formulas in a particular theory. Programs, even of very simple kind, however, often involve multiple data domains, resulting in verification conditions spanning multiple logical theories. To be able to verify high-level programs with rich data types, we have to find decision procedures that can reason on complex domains.

Designing a decision procedure not only requires a good understanding of the algebraic properties of the specific domains, but also requires a lot of effort and often ingenuity to exploit these algebraic properties. Decision procedures for simple domains are well-studied in this way; by exploiting specific structures of the domains, efficient algorithms are obtained whose complexity in most cases matches the optimal theoretical bounds. Designing a decision procedure for a combined theory from scratch, however, is not practical because there are potentially many different combinations. Combined structures, however, usually are constructed in a modular way, by imposing additional constructs on component substructures. Therefore, it is natural to investigate whether decision

* This is a summary of my PhD thesis. All numberings refer to those in the thesis.

procedures for such combined structures can be obtained modularly, namely, by utilizing decision procedures for the component theories as black boxes.

This line of investigation goes back to Nelson and Oppen who discovered a combination procedure for stably infinite quantifier-free theories with disjoint signatures. The procedure combines satisfiability procedures for component theories using equality propagation. The applicability of the Nelson-Oppen method, however, is limited mainly by two conditions: (i) it works only with quantifier-free theories and, (ii) it requires that the signatures of component theories are disjoint. It turns out that these two restrictions are very hard to remove in spite of much research in this direction. There are several recent advances for combining quantified theories or non-disjoint theories. The practical applicability of these general combination methods, however, is limited in that these methods require additional nontrivial conditions on component theories or structures. In fact accumulated evidence shows that the existence of modular combinations like the Nelson-Oppen method should be viewed as an exception. In combining theories with quantifiers or non-disjoint signatures, we should not expect the same level of modularity as in the Nelson-Oppen schema, but rather rely on close examination of the characteristics of the combining constructs as well as the properties of the individual component theories.

**Our Contribution**

This thesis presents another approach to the combination problem; instead of seeking solutions to general purpose combinations, we study specific combination problems by exploiting algebraic properties of the combined domain. As a result, we offer novel solutions to an important class of decision problems which commonly appear in program verification, namely, the mixed constraints on data structures with constraints on "integral measures" of those data structures. Such constraints can express a wide range of program properties, in particular memory safety properties such as absence of memory overflow and out-of-bound array access, which are crucial for program correctness.

The combination consists of recursively defined data structures and Presburger arithmetic integrated by the size function that maps a data object to its size. This kind of combination tightly links data domains with integer domains, rendering a combination that does not fall under the Nelson-Oppen framework nor any other current combination techniques.

Our approach is to reduce constraints on data structures to constraints on integers, and in the presence of quantifiers, to reduce quantifiers on data objects to quantifiers on integers. The technical development divides into three parts.

**Decision Procedures for Term Algebras with Presburger Arithmetic.** Term Algebras can model a variety of tree-like data types such as records, lists, stacks, etc., which are essential constructs in programming languages. We developed the basic reduction technique, namely, extraction of accurate integer constraints from term constraints. From the construction of accurate integer constraints that

precisely characterize term constraints, we can derive decision procedures for the combined constraints by utilizing decision procedures for term algebras and decision procedures for integer arithmetic.

We showed that for structures with infinite constant domain such an accurate integer constraint, which is satisfiable if and only if the corresponding term constraint is satisfiable, can be effectively computed and is expressible by a quantifier-free Presburger formula linear in the size of the term constraint. For structures with finite constant domain we introduce an additional counting constraint to account for the fact that with finitely many constants the number of distinct terms of a particular length is bounded. We showed that also this counting constraint is expressible by a quantifier-free Presburger formula. The latter decision procedure directly extends Oppen's decision procedure for infinite data domains to finite domains.

For the first-order theory, we first developed a new quantifier elimination procedure for the theory of *pure* term algebras which can eliminate blocks of quantifiers of the same kind in one step. Then we extended it to a decision procedure for the theory of term algebras with integer constraints by, again, extracting integer constraints from term constraints combined with a reduction of quantifiers on term variables to quantifiers on integer variables. The complexity of the new decision procedure is $k$-fold exponential for formulas with $k$ quantifier alternations. This is optimal in the sense that the theory of pure term algebras itself has non-elementary complexity.

**Decision Procedures for Queues with Presburger Arithmetic.** A queue is a typical data type of linear structure. It is widely used in programming languages and forms the basis for many concurrent algorithms and communication protocols. As a queue can grow at both ends, it does not fall in the category of recursive data structures, which can be modeled as term algebras. For this reason, we have further improved the reduction technique and developed new normalization procedures to handle the distinguished properties of queues. With these new improvements, we designed decision procedures for the quantifier-free theories as well as quantifier elimination procedures for the first-order theories.

**Decidability of the First-order Theory of Knuth-Bendix Order.** Using quantifier elimination and the reduction technique developed for solving the decision problem of term algebras augmented with Presburger arithmetic, we proved the decidability of the first-order theory of Knuth-Bendix Order, thereby solving a long-standing open problem in term rewriting (officially listed as RTA open problem 99 since 2000).

This decidability result is obtained by quantifier elimination on a complex structure containing term algebras and Presburger arithmetic. In this structure, we have a weight function mapping terms to integers as well as various boundary functions mapping integers to terms. In addition, the Knuth-Bendix order is expanded in two directions. First, the order is decomposed into three disjoint

suborders depending on which of three conditions is used in the definition. Secondly, all orders (including the suborders) are extended to gap orders, which assert the least number of distinct objects between two terms. Moreover, as Knuth-Bendix order is recursively defined on a lexicographic extension of itself, gap orders are extended to tuples of terms. Thus we actually established the decidability of a richer theory. In constructing the quantifier elimination procedure, we overcame several technical challenges, including simplification of complex literals, elimination of integer quantifiers, elimination of equality, elimination of negative literals, and proof of termination.

Knuth-Bendix order has numerous applications in term rewriting and theorem proving, along with *lexicographic path order*. In ordered term rewriting, a strategy built on ordering constraints can dynamically orient an equation, at the time of instantiation, even if the equation is not uniformly orientable. This provides a powerful tool to prove the termination of rewriting systems. In ordered resolution and paramodulation, ordering constraints are used to select maximal literals to perform resolution. It also serves as enabling conditions for inference rules and such conditions can be inherited from previous inferences at each deduction step. This helps to prune redundancy of the search space without compromising refutational completeness. Unfortunately, the first-order theory of lexicographic path order is undecidable. Therefore, our result on the decidability of Knuth-Bendix order may greatly benefit future algorithm design in term rewriting and theorem proving.

It is interesting that the combination of term algebras with integer arithmetic can help solve an open problem in another quite different field. We believe that this demonstrates the effectiveness of our approach to the combination problem; studying concrete combination types where richer algebraic properties can be exploited.

**Thesis Organization and Publications**

Chapter 2 presents decision procedures for the theory of term algebras with integers. It introduces the technique to reduce term constraints to integer constraints, and in the presence of quantifiers, term quantifiers to integer quantifiers. These decision procedures were first published, without proofs, in IJCAR'04 [60] (*Best Paper Award*) and in TPHOLs'04 [61]; an expanded version, including proofs, appears in [64]. The work in Section 2.5 (verifying red-black trees) will appear in [65]. Chapter 3 adapts the reduction technique to construct decision procedures for the theory of queues with integer arithmetic. A part of this work was published in FSTTCS'05 [63]. Chapter 4 presents the proof of decidability of the first-order theory of Knuth-Bendix order using quantifier elimination and the reduction technique developed for the theory of term algebras with Presburger arithmetic. This result was published in CADE'05 [62]. Chapter 5 concludes the thesis with a discussion of future work.

**Preliminaries**

We assume the first-order syntactic notions of variables, parameters and quantifiers, and semantic notions of structures, satisfiability and validity as in [16].

All decision procedures for quantifier-free theories presented in this thesis are *refutation-based*; to determine the validity of a formula $\varphi$, they determine the unsatisfiability of $\neg\,\varphi$, which further reduces to determining the unsatisfiability of each disjunct in the *disjunctive normal form* of $\neg\,\varphi$. Henceforth, in discussions related to quantifier-free theories, a quantifier-free formula always refers to a conjunction of literals.

All decision procedures for the first-order theories presented in this thesis are based on *quantifier elimination*. A theory $T$ is said to *admit quantifier elimination* if any formula can be equivalently (modulo $T$) and effectively transformed into a quantifier-free formula. If a theory admits quantifier elimination, then the truth value of any sentence is reducible to the truth value of a ground formula.

We present algorithms in a nondeterministic manner; whenever we say "guess $\phi$", we mean to add a valid (w.r.t. the context) disjunction $\bigvee_i \phi_i$ (where $\phi$ is one of the disjuncts) to the target formula. When we replace $\phi$ by $\bigvee_i \phi_i$ or directly introduce $\bigvee_i \phi_i$, it should be understood that an implicit disjunctive splitting is carried out and we work on each resultant disjunct "simultaneously".

## Chapter 2. Decision Procedures for Term Algebras with Presburger Arithmetic

In this chapter we present the integration of Presburger arithmetic with term algebras. It first introduces the technical machinery and then presents decision procedures for quantifier-free theories and quantified theories.

Term algebras can represent an important class of recursively defined data structures known as *recursive data structures*. This class of structures satisfies the following two properties of term algebras: (i) the data domain is the set of data objects generated exclusively by applying constructors, and (ii) each data object is uniquely generated. Examples of such structures include lists, stacks, counters, trees and records; queues do not belong to this class as they are not uniquely generated: they can grow at both ends. The combined constraints can express memory safety properties (e.g., absence of memory overflow) and other augmented properties of data structures (e.g., being a balanced tree).

Our language of the integrated theory has two sorts; the integer sort $\mathbb{Z}$ and the term sort $\mathbb{T}$. The language is the set-theoretic union of the language of term algebras and the language of Presburger arithmetic plus the additional integer functions mapping terms to natural numbers. Formulas are formed from *term literals* and *integer literals* in the usual way. Term literals are exactly the literals in the theory of term algebras. Integer literals are those that can be built up from primitive integer terms (the length function applied to $\Sigma$-terms), addition and the other usual arithmetic functions and relations.

Formally we define the structure of term algebras with one integer function as $\mathrm{TA}_{\mathbb{Z}} = \langle \mathrm{TA}; \mathrm{PA}; |\cdot| : \mathbb{T} \to \mathbb{N} \rangle$, where TA is a term algebra, PA is Presburger arithmetic, and $|\cdot|$ is the length function defined recursively by (i) for any constant $a$, $|a| = 1$, and (ii) for a term $\alpha(t_1, \ldots, t_k)$, $|\alpha(t_1, \ldots, t_k)| = 1 + \sum_{i=1}^{k} |t_i|$. The extended language is denoted by $\mathscr{L}_{\mathbb{T}}^{\mathbb{Z}}$. Note that we choose to use the length function just for ease of presentation. Generalizing it into a weight function that assigns an arbitrary nonnegative integer to each symbol, or a height function that gives the length of the maximum path does not require any essential changes to our techniques. In fact we can have more than one length function (see Section 2.5).

We present decision procedures for the quantifier-free and the first-order theory of term algebras with length function and integer constraints, for structures with both finite and infinite constant domain. We will use the following notation for these theories. $\mathrm{Th}^{\forall}(\mathrm{TA})$ and $\mathrm{Th}^{\forall}(\mathrm{TA}_{\mathbb{Z}})$ denote the quantifier-free theory of, respectively, pure term algebras and term algebras with a length function and Presburger arithmetic constraints. Similarly, $\mathrm{Th}(\mathrm{TA})$ and $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$ denote the full first-order theory of pure term algebras and term algebras with a length function and Presburger arithmetic constraints. The decision procedures for $\mathrm{Th}^{\forall}(\mathrm{TA}_{\mathbb{Z}})$ are based on Oppen's algorithm for acyclic recursive data structures with infinite data domain ([45]). To decide satisfiability of a term constraint $\varphi$, Oppen's procedure constructs a DAG for $\varphi$, extracts from this DAG all implied equalities between terms, and then checks for inconsistencies with disequalities in $\varphi$. We extend this procedure to $\mathrm{Th}^{\forall}(\mathrm{TA}_{\mathbb{Z}})$ by extracting an implied *length constraint* from the term constraint. We show that for structures with infinite constant domain such a length constraint, which is satisfiable if and only if the term constraint $\varphi$ is satisfiable, can be effectively computed and is expressible by a quantifier-free Presburger formula linear in the size of $\varphi$. For structures with finite constant domain we introduce an additional *counting constraint* to account for the fact that with finitely many constants the number of distinct terms of a particular length is bounded. We show that also this counting constraint is expressible by a quantifier-free Presburger formula. The latter decision procedure directly extends Oppen's decision procedure for infinite data domains to finite domains. As a case study, we analyze the red-black tree algorithm using $\mathrm{Th}^{\forall}(\mathrm{RB}_{\mathbb{Z}})$, the theory of a term algebra with two integer functions. This serves as an example on how to apply our method to theories of term algebras with more than one integer function. The complexity of all decision problems for the quantifier-free theories present in this chapter is NP-complete.

For the first-order theory, we first present a new quantifier elimination procedure for $\mathrm{Th}(\mathrm{TA})$ and then extend it to an elimination procedure for $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$. Our elimination procedure for $\mathrm{Th}(\mathrm{TA})$ is based on the elimination procedure in [19], but can eliminate blocks of quantifiers of the same kind in one step. We extend it to a decision procedure for $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$ by, again, extracting integer constraints from term constraints combined with a reduction of quantifiers on term variables to quantifiers on integer variables. The complexity of elimination procedures for the first-order theories present in this chapter is $k$-fold exponential for prenex formulas with $k$ quantifier alternations, regardless of the number

of quantifiers. This is optimal in the sense that the theory of pure term algebras itself is non-elementary.

The decision procedures for $\mathrm{Th}^{\forall}(\mathrm{TA}_{\mathbb{Z}})$ and $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$ were first published, without proofs, in [60]. The improved version that allows elimination of blocks of quantifiers was published in [61]. In that paper we showed that the complexity of our decision procedures was $2k$-fold exponential for $k$ quantifier alternations for $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$. The presentation in this chapter bases on [64] which provides an extended presentation of [60, 61], improves the complexity of the decision procedure for $\mathrm{Th}(\mathrm{TA}_{\mathbb{Z}})$ to $k$-fold exponential for $k$ quantifier alternations, and includes all the proofs. The work in Section 2.5 will appear in [65].

### Main Theorem

Consider $\Phi_{\mathbb{T}}$ and $\Phi_{\mathbb{Z}}$ are solvable in their respective theories. It is a simple but crucial observation that $\Phi_{\mathbb{T}} \wedge \Phi_{\mathbb{Z}}$ is solvable if and only if the length constraint $\Phi_{\Delta}$ induced by $\Phi_{\mathbb{T}}$ does not contradict $\Phi_{\mathbb{Z}}$. We call a Presburger formula $\Phi_{\Delta}$ a *length constraint completion* (LCC) of a term constraint $\Phi_{\mathbb{T}}$ if $\Phi_{\Delta}$ exactly characterizes the solution set of $\Phi_{\mathbb{T}}$. The whole thesis work builds on the following central theorem.

**Main Theorem** ([60]). Let $\Phi_{\Delta}$ be an LCC for $\Phi_{\mathbb{T}}$. Then $\mathrm{TA}_{\mathbb{Z}} \models_{\exists} \Phi_{\mathbb{T}} \wedge \Phi_{\mathbb{Z}}$ if and only if $\mathrm{PA} \models_{\exists} \Phi_{\Delta} \wedge \Phi_{\mathbb{Z}}$.

By this theorem the decision problem reduces to the computation of desired LCCs in Presburger arithmetic. In fact we need a refined notion of LCC defined for $\Phi_{\mathbb{T}} \wedge \theta_{\mathbb{Z}}$. (In the thesis, we call it an LCC for $\Phi_{\mathbb{T}}$ *relativized* to $\theta_{\mathbb{Z}}$, or an RLCC for $\Phi_{\mathbb{T}}/\theta_{\mathbb{Z}}$.)

### Decision Procedures for Quantifier-free Theories

We have the following scheme for combining quantifier-free theories.

**Generic Decision Procedure.** Input: $\Phi_{\mathbb{T}} \wedge \Phi_{\mathbb{Z}}$.

1. Return *FAIL* if $\mathrm{TA} \not\models_{\exists} \Phi_{\mathbb{T}}$.
2. For each partition $\Phi_{\mathbb{T}}^{(i)} \wedge \theta_{\mathbb{Z}}^{(i)}$ of $\Phi_{\mathbb{T}}$:
   (a) Compute an LCC $\Phi_{\Delta}^{(i)}$ for $\Phi_{\mathbb{T}}^{(i)} \wedge \theta_{\mathbb{Z}}^{(i)}$.
   (b) Return *SUCCESS* if $\mathrm{PA} \models_{\exists} \Phi_{\Delta}^{(i)} \wedge \Phi_{\mathbb{Z}}$.
3. Return *FAIL*.

For structures with infinite constant domain, an LCC for $\Phi_{\mathbb{T}}$ can be "read off" from the DAG representation of $\Phi_{\mathbb{T}}$. However, for structures with finite constant domain the case is more complicated; LCCs are induced not only by the structures of objects, but also by the size of the constant domain. For this purpose we introduce *counting constraints*. A *counting constraint* is a predicate $\mathrm{CNT}_{k,n}(x)$ ($k > 0, n \geq 0$) that is *true* if and only if there are at least $n+1$ different

terms of length $x$ in TA with a constant domain having $k$ constants. As an example, consider Lisp list structures with nil being the only constant, $\mathrm{CNT}_{1,n}(x)$ is $x \geq 2m-1 \wedge 2 \nmid m$ where $m$ is the least number such that the $m$-th *Catalan number* $C_m = \frac{1}{m}\binom{2m-2}{m-1}$ is greater than $n$. This is not surprising as $C_m$ gives the number of binary trees with $m$ leaves (that tree has $2m-1$ nodes). Counting constraints also play an important role in reducing term constraints to integer constraints in the decidability proof of the first-order theory of Knuth-Bendix order (Chapter 4).

For all quantifier-free theories, we show that the corresponding LCCs can be expressed in Presburger arithmetic and computed in linear time. It follows that all decision problems for quantifier-free theories considered in this chapter are NP-complete.

**Verifying Red-Black Trees**

Our method for deriving decision procedures can be applied to theories with more than one integer function on terms. In Section 2.5 we show a theory of a term algebra with two integer functions to express the properties of red-black trees.

Formally the structure of red-black trees is

$$\mathrm{RB}_{\mathbb{Z}} = \langle\, \mathrm{RB};\, \mathrm{PA};\, |\cdot|_{\max}, |\cdot|_{\min} : \mathbb{T}_{\mathrm{rb}} \to \mathbb{N} \,\rangle\ ,$$

where, RB denotes the term algebra with the domain $\mathbb{T}_{\mathrm{rb}}$ built up by constructors, red and black, and one constant nil; $|x|_{\max}$ (resp. $|x|_{\min}$) gives the maximal (resp. minimal) number of black nodes that $x$ can have on a path. The corresponding language is denoted by $\mathscr{L}^{\mathbb{Z}}_{\mathrm{RB}}$.

In $\mathscr{L}^{\mathbb{Z}}_{\mathrm{RB}}$ the property that $x$ is a red-black tree can be expressed by the conjunction of the following three conditions.

| | | |
|---|---|---|
| (§1) | $\lvert x\rvert_{\max} = \lvert x\rvert_{\min}$ | *any maximal path of x contains the same number of black nodes,* |
| (§2) | $\lvert x\rvert_{\max} > 0$ | *any red node of x must have two black children,* |
| (§3) | $\mathrm{Is}_{\mathrm{black}}(x)$ | *the root of x is black.* |

By combinatorial analysis on tree structures, we show that the corresponding LCCs can be computed in Presburger arithmetic in linear time. It follows that $\mathrm{Th}^{\forall}(\mathrm{RB}_{\mathbb{Z}})$ is NP-complete. In fact by the techniques presented in the subsequent sections, we can show that $\mathrm{Th}(\mathrm{RB}_{\mathbb{Z}})$ admits quantifier elimination and hence is decidable. To the best of our knowledge, it is the first known decidable first-order theory of a balanced tree structure.

**Decision Procedures for Quantified Theories**

We first show a new quantifier elimination procedure for the theory of pure term algebras (Section 2.6). It eliminates a block of quantifiers of the same kind in one step regardless of the length of the block. It follows that the complexity of

this elimination procedure is $k$-fold exponential for formulas with $k$ quantifier alternations.

To get an LCC in the combined theory we need to express in Presburger arithmetic the set of legitimate lengths such that a certain number of distinct terms of any length in the set can co-exist. For quantifier-free theories we used a notion called *equality completion*, which basically is the maximal consistent set of all equality and disequality information. However, in the situation of quantifier elimination, we have to deal with parameters (i.e., universally quantified variables). In general the computation of an equality completion introduces more literals, especially disequalities, which may again destroy the completion because it may cause generation of new terms in the subsequent operations. To avoid compromising convergence, we introduce the notion of *clusters* which is weaker than equality completion but contains sufficient information to allow extracting counting constraints.

To construct an LCC for $\Phi_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y})$ (where $\boldsymbol{y}$ denote parameters), we require that $\Phi_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y})$ be cluster complete and in *strongly solved form*, that is, $\Phi_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y})$ is solved in $\boldsymbol{x}$ and all literals of the form $Ly \neq t(\boldsymbol{x}, \boldsymbol{y})$, where $y \in \boldsymbol{y}$ and $t(\boldsymbol{x}, \boldsymbol{y})$ is a constructor term (properly) containing $\boldsymbol{x}$, are redundant. We say that $(\exists \boldsymbol{x} : \mathbb{T})[\Phi_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y})]$ is in *strongly solved form* if $\Phi_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y})$ is strongly solved in $\boldsymbol{x}$.

We outline the key step of this elimination procedure. After a sequence of normalization and removal of redundant disequalities, we obtain

$$(\exists \boldsymbol{x} : \mathbb{T}) \Big[ \Phi^{(1)}_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \ \wedge \ \Phi^{(2)}_{\mathbb{T}}(\boldsymbol{y}) \ \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y}) \ \wedge \ \Phi_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \Big] \ ,$$

which is in strongly solved form and cluster complete. We compute an LCC $\Phi_{\Delta}(\boldsymbol{x}, \boldsymbol{y})$ for $\Phi^{(1)}_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \wedge \Phi^{(2)}_{\mathbb{T}}(\boldsymbol{y}) \wedge \theta_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y})$, producing the equivalent

$$(\exists \boldsymbol{x} : \mathbb{T}) \Big[ \Phi^{(1)}_{\mathbb{T}}(\boldsymbol{x}, \boldsymbol{y}) \ \wedge \ \Phi^{(2)}_{\mathbb{T}}(\boldsymbol{y}) \ \wedge \ \Phi_{\Delta}(\boldsymbol{x}, \boldsymbol{y}) \ \wedge \ \Phi_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \Big] \ ,$$

which reduces to

$$\Phi^{(2)}_{\mathbb{T}}(\boldsymbol{y}) \ \wedge \ (\exists \boldsymbol{u} : \mathbb{Z}) \Big[ \Phi_{\Delta}(\boldsymbol{u}, \boldsymbol{y}) \ \wedge \ \Phi_{\mathbb{Z}}(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{z}) \Big] \ .$$

Here a block of existential term quantifiers is transformed into a block of existential integer quantifiers.

**Related Work and Comparison.**

Our component theories are both decidable. Presburger arithmetic was first shown to be decidable in 1929 by quantifier elimination [16]. A more efficient algorithm was later discovered by Cooper [12] and further improved by Reddy and Loveland [46]. It is well-known that recursive data structures can be modeled as term algebras which were shown to be decidable by Malcev using quantifier elimination [36]. This result was proved again several times in different settings [34, 9, 19, 8, 3, 49, 30, 29, 60]. Quantifier elimination has been used to obtain decidability results for various extensions of term algebras. Maher showed

the decidability of the theory of infinite and rational trees [34]. Comon and Delor presented an elimination procedure for term algebras with membership predicate in the regular tree language [8]. Backofen presented an elimination procedure for structures of feature trees with arity constraints [3]. Rybina and Voronkov showed the decidability of term algebras with queues [49]. Kuncak and Rinard showed the decidability of term powers, which are term algebras augmented with coordinate-wise defined predicates [30]. Decision procedures for the quantifier-free theory of recursive data structures were discovered by Nelson, Oppen et al. [40, 45, 15]. Oppen gave a linear algorithm for acyclic structures [45] and (with Nelson) a quadratic algorithm for cyclic structures [40]. If the values of the selector functions on constants are specified, then the problem is NP-complete [45].

A general combination method for decision procedures for quantifier-free theories was developed by Nelson and Oppen in 1979 [39]. The method requires that component theories be loosely coupled, that is, have disjoint signatures, and are stably infinite[1] [53]. Tinelli and Ringeissen presented a general theoretical framework for combining satisfiability procedures of theories with non-disjoint signatures [54]. Tinelli and Zarba generalized Nelson-Oppen's method to theories in multisorted languages [55]. Armando, Ranise and Rusinowitch presented a uniform framework using superposition for deriving decision procedures for certain combined theories [1]. Ghilardi presented a set of model-theoretical conditions for the existence of Nelson-Oppen combination schema on theories having non-disjoint signatures [18]. But none of these general purpose combination methods are applicable to the combination of our component theories, which is a multisorted theory with a function mapping elements in one sort to another.

Zarba constructed decision procedures for a combined theory of sets and integers and a theory of multisets and integers, respectively [59, 58]. The integration of Presburger arithmetic with recursive data structures was discussed by Bjørner [5] and an incomplete procedure was implemented in STeP (Stanford Temporal Prover) [6].

Integer constraints not only arise in the combination of decision procedures, but they are also useful as an auxiliary extension to encode properties on data structures. This line of investigation goes back to Skolem who showed the decidability of the first-order theory of Boolean algebras by reducing constraints on sets to constraints on the cardinality of sets [51]. It readily follows from the reduction technique that the first order theory of sets with cardinality constraints in Presburger arithmetic is decidable [17]. Recently, Revesz [47], and Kuncak and Rinard [31] independently presented decision procedures for this theory. A combination of Presburger arithmetic and term algebras was used by Korovin and Voronkov to show that the quantifier-free theory of term algebras with Knuth-Bendix order is NP-complete [26, 27]. Along this line of investigation we proved the decidability of the first-order theory of Knuth-Bendix orders [62]

[1] A theory is stably infinite if a quantifier-free formula in the theory is satisfiable if and only if it is satisfiable in an infinite model.

using quantifier elimination (Chapter 4). The elimination procedure makes extensive use of Presburger arithmetic in the reduction of quantifiers on term variables to quantifiers on integer variables.

## Chapter 3. Decision Procedures for Queues with Presburger Arithmetic

In this chapter we present the integration of Presburger arithmetic with queues. It first adapts the technical machinery introduced in Chapter 2 and then presents decision procedures for quantifier-free theories and quantified theories.

Queues are widely used in many fields of computer science such as communication networks, job scheduling and simulation. They also provide an important synchronization mechanism in modeling distributed protocols and hence form the basis of many concurrent algorithms.

As in the case of term algebras, we use a multi-sorted language which has three sorts: atoms ($\mathcal{A}$), integers ($\mathbb{Z}$), and queues ($\mathcal{Q}$). The language is the set-theoretic union of the language of queues and the language of Presburger arithmetic connected by the length function $|\cdot| : \mathcal{Q} \to \mathbb{N}$. It allows us to express semantics of string operations in the C language. For example, `strncmp` can be expressed in the existential theory of queues with prefix relation and Presburger arithmetic as follows.

$$
\begin{array}{ll}
 & \text{strncmp(const char } *s_1\text{, const char } *s_2\text{, size\_t n)} : \\
 & res = 0 \ \wedge \ \exists q\,(\,|q| = n \ \wedge \ q \preceq s_1 \ \wedge \ q \preceq s_2\,) \\
\vee & res > 0 \ \wedge \ \Big[\,(s_1 \neq s_2 \ \wedge \ s_2 \preceq s_1 \ \wedge \ |s_2| \leq n) \\
 & \qquad\qquad \vee \ \exists q \Big( \bigvee_{c<c'} (q \circ c' \preceq s_1 \ \wedge \ q \circ c \preceq s_2 \ \wedge \ |q| < n) \Big) \,\Big] \\
\vee & res < 0 \ \wedge \ \Big[\,(s_1 \neq s_2 \ \wedge \ s_1 \preceq s_2 \ \wedge \ |s_1| \leq n) \\
 & \qquad\qquad \vee \ \exists q \Big( \bigvee_{c<c'} (q \circ c' \preceq s_2 \ \wedge \ q \circ c \preceq s_1 \ \wedge \ |q| < n) \Big) \,\Big] \ .
\end{array}
$$

We present decision procedures for quantifier-free theories of queues with Presburger arithmetic. We consider two kinds of quantifier-free theories, based on whether they include the prefix relation or not. We also present a quantifier elimination procedure for the first-order theory of queues with integers. The elimination procedure removes a block of existential quantifiers in one step. In all developments, we assume that the atom domain is finite; the decision problems in an infinite domain are considerably easier.

In Chapter 2 we gave decision procedures for the theory of term algebras with integer constraints. The method relies on a key normalization process to extract integer constraints from term constraints. The normalization partitions terms into stratified clusters such that (1) each cluster consists of pairwise unequal terms (trees) of the same length, and (2) disequalities between composite

terms (proper trees) in a cluster are implied by disequalities in the clusters of lower ranks. Property (2) allows the construction of a satisfying assignment in a bottom-up fashion, while providing integer constraints that precisely characterize the satisfiability of the clusters. Thus, (1) and (2) allow us to reduce the satisfiability of the original formula to the satisfiability of computable integer constraints. The decision procedures presented in this chapter rely on the same idea. But for queues, disequalities cannot be normalized into stratified clusters, because queues are not uniquely generated (they can grow at both ends). Consider, for example, the constraint

$$X \neq Y \ \wedge \ aX \neq Yb \ \wedge \ Xa \neq bY \ \wedge \ |X| = |Y| \ .$$

Clearly infinitely many assignments of the form $\{X = (ba)^n b, Y = a(ba)^n\}$ satisfy $X \neq Y$, but neither $aX \neq Yb$ nor $Xa \neq bY$. As a consequence, we cannot construct a satisfying assignment inductively. In this chapter we present new normalization procedures that allow the computation of a *cut length* $L_t$ for all queue variables: below $L_t$ all satisfying assignments can be enumerated; above $L_t$ integer constraints can be computed that are equisatisfiable with the original formula.

The decision procedures for $\mathrm{Th}^{\forall}(\mathfrak{Q}_{\mathbb{Z}})$ and $\mathrm{Th}(\mathfrak{Q}_{\mathbb{Z}})$ were first published, without proofs, in [63]. This chapter provides an extended presentation of [63], presents a decision procedure for $\mathrm{Th}^{\forall}(\mathfrak{Q}^{+}_{\mathbb{Z}})$, the quantifier-free theory with the prefix predicate, and includes all the proofs.

### Technical Machinery

Given a quantifier-free formula $\Phi_{\mathcal{Q}} \wedge \theta_{\mathbb{Z}}$, we first compute $\Phi_{\Delta+}$, an over-approximation of the LCC for $\Phi_{\mathcal{Q}} \wedge \theta_{\mathbb{Z}}$. $\Phi_{\Delta+}$ over-approximates the LCC in the sense that any satisfying assignment for $\Phi_{\mathcal{Q}} \wedge \theta_{\mathbb{Z}}$ also satisfies $\Phi_{\Delta+}$, but not necessarily vice versa. The next goal is to narrow down $\Phi_{\Delta+}$ to $\Phi_{\Delta}$ so that for any satisfying assignment for $\Phi_{\Delta}$, there is a satisfying assignment for $\Phi_{\mathcal{Q}} \wedge \theta_{\mathbb{Z}}$.

It is a simple observation that if $\Phi_{\mathcal{Q}}$ is satisfiable, then it can be satisfied by sufficiently long queues: there exists a *cut length* $\delta$ such that if $\mathfrak{Q} \models_{\exists} \Phi_{\mathcal{Q}}$, then for any solution $(l_i)_n$ (i. e., $l_0, \ldots, l_n$) for $\Phi_{\Delta+}$ such that $l_i \geq \delta$, there exists a solution $(\alpha_i)_n$ for $\Phi_{\mathcal{Q}}$ such that $|\alpha_i| = l_i$. Let $C_{\Phi}(\delta)$ denote $\bigwedge_{X \in \mathcal{V}_{\mathcal{Q}}(\Phi_{\mathcal{Q}})} |X| \geq \delta$, where $\mathcal{V}_{\mathcal{Q}}(\Phi_{\mathcal{Q}})$ denotes the set of variables of sort $\mathcal{Q}$ and appearing in $\Phi_{\mathcal{Q}}$. It is easily seen that $\Phi_{\Delta+} \wedge C_{\Phi}(\delta) \wedge \theta_{\mathbb{Z}}$ is an LCC for $\Phi_{\mathcal{Q}} \wedge C_{\Phi}(\delta) \wedge \theta_{\mathbb{Z}}$. So computing LCCs reduces to computing an upper bound of $\delta$. However, there exist anomalies in which $\delta$ is not the smallest $max\{(\mu_i)_n\}$ such that

$$\mathfrak{Q}_{\mathbb{Z}} \models_{\exists} \Phi_{\mathcal{Q}} \ \wedge \ \bigwedge_{0<i\leq n} |X_i| = \mu_i \ ,$$

where $(X_i)_n$ enumerate $\mathcal{V}_{\mathcal{Q}}(\Phi_{\mathcal{Q}})$. To avoid anomalies we separate the search for a satisfying assignment into two cases. We compute a cut length $L_t \geq \delta$ and enumerate all assignments $\sigma$ with $|[\![X]\!]\sigma| < L_t$, while for $|[\![X]\!]\sigma| \geq L_t$ the

satisfiability of the queue constraints is reduced to the satisfiability of integer constraints in the same way as in Chapter 2.

The computation of $L_t$ is based on the observation that an assignment $\sigma$ is satisfying if every $[\![X]\!]\sigma$ includes a unique "marker" at the same, fixed, position. Such a marker can be constructed by concatenating a "shortest unused prefix", called delimiter, and a unique identifier (for each queue variable and proper constant queue), called color. Let $L_p$ be the length of a shortest delimiter (there can be more than one). Let $L_c$ be the length of colors (all colors are of the same length). We show that $L_c + L_p = L_t \geq \delta$. We call a *length configuration* the conjunction $\bigwedge_{X \in \mathcal{V}_Q(\Phi_Q)} A_X$ where $A_X$ is either $|X| = i$ (for some $i < L_t$) or $|X| \geq L_t$. The purpose of $C$ is to "refine" the over-approximation of $\Phi_{\Delta+}$. It can be shown that $\Phi_{\Delta+} \wedge C \wedge \theta_{\mathbb{Z}}$ is an LCC for $\Phi_Q \wedge C \wedge \theta_{\mathbb{Z}}$. A partial assignment $\partial$ is *compatible* with a configuration $C$ if for any variable $X$, $[\![X]\!]\partial$ is defined if and only if $|X| = i$ (for some $i < L_t$) occurs in $C$. We have

**Generic Decision Procedure for $\mathfrak{Q}_{\mathbb{Z}}$.** Input: $\Phi_Q \wedge \theta_{\mathbb{Z}} \wedge \Phi_{\mathbb{Z}}$.

1. For each $C \in \mathcal{C}$,
   (a) Guess a satisfying $\partial$ compatible with $C$ and update $C$, $\Phi_{\Delta+}$, $\theta_{\mathbb{Z}}$ and $\Phi_{\mathbb{Z}}$ accordingly.
   (b) If succeed, return *SUCCESS* if PA $\models_\exists C \wedge \Phi_{\Delta+} \wedge \theta_{\mathbb{Z}} \wedge \Phi_{\mathbb{Z}}$.
2. Return *FAIL*.

Section 3.4 presents a decision procedure for $\mathrm{Th}^\forall(\mathfrak{Q}_{\mathbb{Z}})$. Section 3.5 presents a decision procedure for $\mathrm{Th}^\forall(\mathfrak{Q}^+_{\mathbb{Z}})$, the quantifier-free theory with the prefix predicate. It involves sophisticated normalization procedures for the computation of length configurations. Section 3.6 presents a quantifier elimination procedure for $\mathrm{Th}(\mathfrak{Q}_{\mathbb{Z}})$, which can remove a block of quantifiers of the same kind in one step.

**Related Work and Comparison.**

Bjørner gave a decision procedure for the quantifier-free theory of queues with subsequence relations which consist of prefix, suffix and sub-queue relations [5]. Bjørner also discussed the integer combination for the case of infinite atom domain without the subsequence relations. The quantifier elimination and the complexity of the first-order theory of queues were given by [48] and [50], respectively. By a standard encoding (in which a queue is represented as sets of natural numbers), the first order theory of queues with prefix relation can be interpreted by WS1S, and hence it is decidable. This theory also admits quantifier elimination [4]. Thomas studied theories of words with "equal length" predicate which can be viewed as special integer constraints [52].

Recently Klaedtke and Ruess showed the decidability of a fragment of $WS1S$ with cardinality constraints ($WS1S^{\mathsf{card}}$) and the undecidability of $WS1S^{\mathsf{card}}$ for certain fragments with second-order quantifier alternation [24]. By the standard encoding, the first-order theory of queues with prefix relation and integers

can be interpreted in $WS1S^{\mathsf{card}}$. In particular, the quantifier-free fragment can be interpreted in the decidable fragment of $WS1S^{\mathsf{card}}$ which does not contain alternation of second-order quantifiers on variables occurring in cardinality constraints. Though interpretation in general renders elegant decidability results, it produces less efficient decision procedures in practice, especially if the host theory has high complexity (in this case even the existential $WS1S$ is non-elementary). Moreover, it is unlikely that any interpretation can put the full first-order theory of queues with integer arithmetic into a decidable fragment of $WS1S^{\mathsf{card}}$.

## Chapter 4. Decidability of the First-order Theory of Knuth-Bendix Order

In this chapter we present the decidability proof of the first-order theory of Knuth-Bendix order by quantifier elimination. This is the most important contribution of this thesis.

Two kinds of orderings are widely used in term rewriting and theorem proving. One is *recursive path ordering* (RPO) which is based on syntactic precedence [14]. The other is *Knuth-Bendix ordering* (KBO, [25, 2]) which is of hybrid nature; it relies on numerical values assigned to symbols as well as syntactic precedence [25]. In ordered term rewriting, a strategy built on ordering constraints can dynamically orient an equation, at the time of instantiation, even if the equation is not uniformly orientable. This provides a powerful tool to prove the termination of rewriting systems [10]. In ordered resolution and paramodulation, ordering constraints are used to select maximal literals to perform resolution. It also serves as enabling conditions for inference rules and such conditions can be inherited from previous inferences at each deduction step. This helps to prune redundancy of the search space without compromising refutational completeness [43].

Solving ordering constraints therefore has fundamental importance to ordered rewriting and ordered resolution. The decision procedures for quantifier-free constraints of both types of orderings have been well-studied [7, 21, 41, 38, 42, 26, 27]. However, situations arise where we need to decide the truth values of quantified formulas on those orderings, especially in the $\forall^*\exists^*$ fragment. Examples include checking the soundness of simplification rules in constrained deduction. Consider a "total simplification scheme" given in [23, 11].

$$\frac{s \;\rightarrow\; t \mid c}{s[v]_p \;\rightarrow\; t \mid (c \wedge c' \wedge s|_p = u)} \quad (u \;\rightarrow\; v \mid c') \;\;,$$

where $s|_p$ denotes the subterm occurring at position $p$ in $s$ and $s[v]_p$ denotes the term obtained from $s$ by substituting $v$ for $s|_p$, states that $s \rightarrow t \mid c$ is simplified to $s[v]_p \rightarrow t \mid (c \wedge c' \wedge s|_p = u)$ by $u \rightarrow v \mid c'$ provided *for all* assignments for variables in $s$ which satisfies $c$, *there exists* an assignment for variables in $u$ which satisfies

$c'$ and $s|_p = u$. The soundness of this rule is formally expressed as

$$\mathrm{TA} \models \forall \mathcal{V}(s) \exists \mathcal{V}(u) \Big[\ c \ \rightarrow\ (c' \ \wedge\ s|_p = u)\ \Big]\ ,$$

which necessarily involves quantifier alternation. To determine the soundness of such simplification rules, we need to be able to reason in the $\forall^*\exists^*$ fragment.

Unfortunately, the full first-order theory of lexicographic path ordering (LPO), the most popular form of RPO, is undecidable [56, 11] except for the special case where the language only has unary functions and the precedence order is total [37]. Until now it has been an open question whether the first-order theory of Knuth-Bendix order is decidable (RTA open problem ♯99). In this chapter we answer this question affirmatively by showing that an extended theory of term algebras with Knuth-Bendix order admits quantifier elimination.

The basic framework is the combination of term algebras with Presburger arithmetic. The combination is more tightly coupled than $\mathrm{TA}_{\mathbb{Z}}$ presented in Chapter 2: not only do we have a *weight function* mapping terms to integers, but we also have various *boundary functions* mapping integers to terms. In addition, the Knuth-Bendix order is expanded in two directions. First, the order is decomposed into three disjoint suborders depending on which of three conditions is used in the definition. Secondly, all orders (including the suborders) are extended to gap orders, which assert the least number of distinct objects between two terms. Moreover, as Knuth-Bendix order is recursively defined on a lexicographic extension of itself, gap orders are extended to tuples of terms. Thus we actually establish the decidability of a richer theory.

**Proof Plan**

The quantifier elimination procedure relies on the following two ideas: *solved form* and *depth reduction*.

*Solved Form.* A quantifier-free formula $\varphi(x, \boldsymbol{y})$ is *solved* in $x$ if it is in the form

$$\bigwedge_{i \le m} u_i \prec^{\mathrm{kb}} x \ \wedge\ \bigwedge_{j \le n} x \prec^{\mathrm{kb}} v_j \ \wedge\ \varphi'(\boldsymbol{y})\ ,$$

where $x$ does not appear in $u_i$, $v_i$ and $\varphi'$. It is not hard to argue that $(\exists x)\ \varphi(x, \boldsymbol{y})$ simplifies to

$$\bigwedge_{i \le m, j \le n} u_i \prec_2^{\mathrm{kb}} v_j \ \wedge\ \varphi'(\boldsymbol{y})\ ,$$

where $\prec_n^{\mathrm{kb}}$, called *gap order*, is an extension of $\prec^{\mathrm{kb}}$ such that $x \prec_n^{\mathrm{kb}} y$ states there is an increasing chain from $x$ to $y$ with at least $n - 1$ elements in between [16, page 196]. It is clear that the elimination of $\exists x$ becomes straightforward once the matrix $\varphi(x, \boldsymbol{y})$ is solved in $x$, or equivalently, $\mathrm{depth}^{\varphi}(x) = 0$ (where $\mathrm{depth}^{\varphi}(x)$, called the *depth* of $x$ in $\varphi$, is the maximal length of selector sequences appearing in front of $x$ in $\varphi$). That leads us to the notion of *depth reduction*.

*Depth Reduction.* Let us first consider the simple case where $x$'s outmost function symbol is a proper constructor $\alpha$ and all occurrences of $x$ have depth greater than 0. By introducing new variables $x_1, \ldots, x_{\mathrm{ar}(\alpha)}$ (called the *descendants* of $x$) to represent $x$, we can rewrite $\exists x \varphi(x, \boldsymbol{y})$ to

$$\exists x_1, \ldots, \exists x_{\mathrm{ar}(\alpha)}\, \varphi'(x_1, \ldots, x_{\mathrm{ar}(\alpha)}, \boldsymbol{y}) \ ,$$

where $\mathrm{ar}(\alpha)$ denotes the arity of $\alpha$, and $\varphi'(x_1, \ldots, x_{\mathrm{ar}(\alpha)}, \boldsymbol{y})$ is obtained from $\varphi(x, \boldsymbol{y})$ by substituting $x_i$ for $\mathsf{s}_i^\alpha x$ $(0 < i \leq \mathrm{ar}(\alpha))$, the immediate subterms of $x$. It is clear that $\mathrm{depth}^{\varphi'}(x_i) < \mathrm{depth}^{\varphi}(x)$. If all occurrences of $x$ have the same depth, then by repeating the process we can generate a formula solved in $\boldsymbol{x}^*$ where $\boldsymbol{x}^*$ are descendants of $x$. A difficulty arises when not all occurrences of $x$ have equal depth. So eventually we meet the situation where some occurrences of $x$ have depth 0 and some do not. Here we have to represent all occurrences of $x$ of depth 0 in terms of $\mathsf{s}_1^\alpha(x), \ldots, \mathsf{s}_{\mathrm{ar}(\alpha)}^\alpha(x)$. This amounts to reducing literals of the form $x \prec_n^{\mathsf{kb}} t$ and literals of the form $t \prec_n^{\mathsf{kb}} x$ to quantifier-free formulas using $\mathsf{s}_1^\alpha(x), \ldots, \mathsf{s}_{\mathrm{ar}(\alpha)}^\alpha(x)$. After that we can introduce new variables and do quantifier manipulation just as in the simple case. Therefore by the depth reduction of $x$, we actually mean reducing the depths of the descendants of $x$, and this essentially depends on the reduction of $x \prec_n^{\mathsf{kb}} t$ and $t \prec_n^{\mathsf{kb}} x$. In order to carry out the reduction we need to extend the language extensively.

**Language Extension**

*Decomposition of KBO.* A Knuth-Bendix order $\prec^{\mathsf{kb}}$ can be decomposed into three *disjoint* orders, a *weight order* $\prec^{\mathsf{w}}$, a *precedence order* $\prec^{\mathsf{p}}$, and a *lexicographic order* $\prec^{\mathsf{l}}$:

$$\begin{aligned}
u \prec^{\mathsf{w}} v &\leftrightarrow \mathsf{w}(u) < \mathsf{w}(v) \ , \\
u \prec^{\mathsf{p}} v &\leftrightarrow \mathsf{w}(u) = \mathsf{w}(v) \ \wedge \ \mathrm{type}(u) \prec^{\Sigma} \mathrm{type}(v) \ , \\
u \prec^{\mathsf{l}} v &\leftrightarrow \mathsf{w}(u) = \mathsf{w}(v) \ \wedge \ \mathrm{type}(u) = \mathrm{type}(v) \ \wedge \ u \prec^{\mathsf{kb}} v \ ,
\end{aligned}$$

where $\mathsf{w}(x)$ denotes the weight of $x$ and $\mathrm{type}(x)$ denotes the outmost function symbol of $x$.

*Gap Orders.* To express formulas of the form $\exists x(u \prec^\sharp x \prec^\sharp v)$ $(\sharp \in \{\mathsf{kb}, \mathsf{w}, \mathsf{p}, \mathsf{l}\})$, in a quantifier-free language we need to extend all aforementioned orders to *gap orders* $\prec_n^\sharp$. A gap order $u \prec_n^\sharp v$ $(n > 0)$ states that "$u$ is less than $v$ w.r.t. $\prec^\sharp$, and there are *at least* $n-1$ elements in between." Similarly, $u \preceq_n^\sharp v$ $(n > 0)$ states that "$u$ is less than $v$ w.r.t. $\prec^\sharp$, and there are *exactly* $n-1$ elements in between".

*Boundary Functions.* Consider the formula $u \preceq_1^{\mathsf{w}} v$. Intuitively it states "the weight of $u$ is less than the weight of $v$ and there are no terms $z$ such that $u \prec^{\mathsf{kb}} z \prec^{\mathsf{kb}} v$, that is, $u$ is the largest term of its weight and $v$ is the smallest term of its weight". To express this we introduce *boundary functions*:

1. $0^{\mathsf{w}} : \mathbb{N} \to \mathbb{T}$ such that $0^{\mathsf{w}}(n)$ is the smallest term (w.r.t. $\prec^{\mathrm{kb}}$) of weight $n$,
2. $0^{\mathsf{p}} : \mathbb{N}^2 \to \mathbb{T}$ such that $0^{\mathsf{p}}(n, p)$ is the smallest term (w.r.t. $\prec^{\mathrm{kb}}$) of weight $n$ and type $\alpha_p$,
3. $1^{\mathsf{w}} : \mathbb{N} \to \mathbb{T}$ such that $1^{\mathsf{w}}(n)$ is the largest term (w.r.t. $\prec^{\mathrm{kb}}$) of weight $n$,
4. $1^{\mathsf{p}} : \mathbb{N}^2 \to \mathbb{T}$ such that $1^{\mathsf{p}}(n, p)$ is the largest term (w.r.t. $\prec^{\mathrm{kb}}$) of weight $n$ and type $\alpha_p$,

where, for all of the above, $f(n) = \bot$ and $f(n, p) = \bot$, if no such term exists.

*Extensions to Tuples.* The reduction of literals like $x \prec_n^{\mathrm{kb}} t$ or $t \prec_n^{\mathrm{kb}} x$ eventually comes down to resolving relations between two terms of the same weight and the same outmost function symbol. So we need to extend all aforementioned notions to tuples of terms of the same total weight.

We denote the structure of term algebras with KBO, extended with gap orders, boundary functions and Presburger arithmetic, by

$$\mathrm{TA}^{\mathbb{Z}}_{\mathrm{kb}^+} = \langle \mathrm{TA}_{\mathrm{kb}};\ \mathrm{TA}_{\mathbb{Z}};\ \prec^{\sharp}_n, \preceq^{\sharp}_n,\ \sharp \in \{\mathsf{kb}, \mathsf{w}, \mathsf{p}, \mathsf{l}\}, n \geq 0;\ 0^*_{(\ldots)}, 1^*_{(\ldots)},\ * \in \{\mathsf{w}, \mathsf{p}\} \rangle\ ,$$

and the corresponding language by $\mathscr{L}^{\mathbb{Z}}_{\mathrm{kb}^+}$.

### Key Elimination Procedure

The key elimination procedure is the one that eliminates term quantifiers. We show below its high-level control-flow.

Input: $(\exists \boldsymbol{x} : \mathbb{T})\, [\, \varphi_{\mathrm{kb}^+}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \ \wedge\ \varphi_{\mathbb{Z}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\, ]$.
**while** $\boldsymbol{x} \neq \emptyset$ **do**
  **if** $(\forall x \in \boldsymbol{x})\, \mathrm{depth}^{\varphi_{\mathrm{kb}^+}}(x) > 0$ **then**
    *Depth Reduction*
      Variable Selection
      Decomposition
      Simplification
  **else** $\{(\exists x \in \boldsymbol{x})\, \mathrm{depth}^{\varphi_{\mathrm{kb}^+}}(x) = 0\}$
    *Elimination*
  **end if**
**end while**

It is easily seen that this procedure is a greedy algorithm in the sense that it tries to do *Elimination* as soon as the elimination condition $(\exists x \in \boldsymbol{x})\, \mathrm{depth}^{\varphi_{\mathrm{kb}^+}}(x) = 0$ holds, that is, all term occurrences of $x$ are of depth 0. Otherwise, the algorithm tries to create the elimination condition using *Depth Reduction* which includes three sequential sub-procedures: Variable Selection, Decomposition and Simplification. We require that Variable Selection be done in depth-first manner. As all depths are finite, this guarantees that a run eventually leaves *Depth Reduction* and enters *Elimination*.

### Technical Difficulties

We overcome several technical difficulties to obtain the decidability proof.

*Term Simplification.* In principle, boundary terms can appear in the weight function or in selectors, selector terms can occur in the weight function, and the weight function can be used to construct boundary terms. Repeating this process we can build more and more complex terms. Lemma 4.3 eliminates this superficial complication.

*Depth Reduction.* To reduce the depth of a variable to 0, we need to express equality literals and gap order literals like $u \bowtie v$ in formulas only containing proper subterms of $u$ and $v$. Due to the introduction of boundary terms and gap orders, we have a total of 285 combinations to consider (reductions 4.59-4.344). In the proofs of Lemmas 4.4-4.7, we list the most typical and sophisticated reductions.

*Termination.* The argument for termination is quite subtle. First the depth reduction of a variable may be at the expense of increasing the depth of a term on the other side of a predicate. Moreover, the depth reduction in general introduces more existential quantifiers and more literals in one of resulting formulas. A priori it is a surprise that a special type of order literal, called the open gap order literal, plays a key role in the termination proof. In every step of the transformation the number of open gap order literals in each resulting formula is no more than that in the original formula. Moreover, the final elimination procedure removes at least one open gap order literal if the eliminated variable occurs in such literals. When all open gap order literals are gone, the depths of terms will be strictly decreasing. The detailed argument is given in the proof of Lemma 4.11.

**Related Work and Comparison.**

The decidability of the theory of RPO has been well-studied. Comon proves the decidability of the quantifier-free theory of total lexicographic path ordering (LPO, a variant of RPO) [7]. A similar result holds for RPO [21]. Nieuwenhuis establishes the NP-completeness for the quantifier-free theory of LPO [41]. Narendran, Rusinowitch and Verma obtain a similar result for RPO [38]. A more efficient algorithm for the quantifier-free theory of RPO is given by Nieuwenhuis and Rivero [42]. Comon and Treinen show the undecidability of the first-order theory of LPO and the undecidability of the first-order theory of RPO in case of partial precedence [56, 11]. The decidability of the first-order theory of RPO (LPO) in case of unary signature and total precedence is due to Narendran and Rusinowitch [37]. The decidability of the first-order theory of RPO in case of total precedence remains open.

Recently some partial decidability results for the theory of KBO have been obtained. Korovin and Voronkov show the decidability of the quantifier-free theory of term algebras with KBO [26]. They later improve the algorithm and show that the quantifier-free theory of KBO is NP-complete [27]. Analogous to [37], they also show the decidability of the first-order theory of KBO in the case where all functions are unary [28].

## Chapter 5. Conclusion

This thesis offers novel solutions to an important class of decision problems, the mixed constraints on data structures with quantitative properties. We developed the reduction technique, namely, extraction of accurate integer constraints from data constraints, and in case of quantified theories, reduction of quantifiers on data objects to quantifiers on integers. From the construction of accurate integer constraints that precisely characterize data constraints, we can derive decision procedures for the combined constraints by utilizing decision procedures for data structures and decision procedures for Presburger arithmetic. We presented decision procedures for term algebras with integers and decision procedures for queues with integers. Using our reduction technique and quantifier elimination, we proved the decidability of the first-order theory of *Knuth-Bendix Order*, thereby solving a long-standing open problem in term rewriting (officially listed as RTA open problem 99 since 2000).

We envisage that decision procedures will play a bigger role in formal methods, model checking and program analysis. They will render more valuable tools for specifying and analyzing security applications, and embedded and reactive systems. We plan to expand the thesis work in the following directions.

Security Verification. A vast majority of security problems of software systems is caused by memory access violations such as stack or heap overflow and out-of-bound array access. This brings unprecedented demands in reasoning about *memory safety* properties, that is, memory accesses, in terms of various data manipulations, always stay within designated boundaries. In fact, memory safety properties are a subclass of the more general *quantitative* properties of *resource reallocation* which can be expressed in the language of data structures with integer constraints. We believe work in this thesis can be used as the basis for specifying and verifying such quantitative properties.

High-level Static Analysis. Many advanced data structures are widely used in industry-sized applications such as Java Runtime Library and C++ Standard Template Library. They include linked lists, heaps, priority queues, hash tables, skip lists, splay trees, etc. Program reliability and efficiency rely on *high-level* properties of these data structures. The traditional *low-level* logic representation of these structures easily leads to undecidability. Here the challenge is to strike the right balance between expressive power and complexity. A specification language should be well-designed so that it can model the *core* properties of a data structure while retaining decidability or even low complexity. We believe more new decision procedures for advanced data structures will make important contributions to high-level static analysis.

Verification of Embedded and Reactive Systems. It is of essential importance to develop techniques for designing and analyzing *embedded* and *reactive* systems, as they are ubiquitous in our daily lives, particularly in many safety-critical devices that we use. One of the challenges that we would like to focus on is

to carry out symbolic exploration of the state-space efficiently. Such symbolic computation unavoidably involves quantified formulas, while many first-order theories either are undecidable or intractable due to high complexity. As many have observed, however, we hardly deal with formulas with a large quantifier alternation depth, and hence it is worthwhile to investigate the class of formulas that can have arbitrarily long sequences of quantifiers of the same kind while the total number of quantifier alternations is bounded by a constant number. In the search of quantifier elimination procedures for the theory of term algebras with integers and for the theory of queues with integers, we already aimed at and successfully obtained *block-wise* quantifier eliminations which are practically more efficient than *single-variable* quantifier eliminations. We propose to continue the development of more efficient quantifier eliminations, in particular elimination procedures for the combined theory of integer and real arithmetic, which finds applications in the verification of hybrid systems and real-time systems.

More Powerful Decision Theories. On the theoretical front, we plan to search for new powerful tools to prove decision problems. Currently we are investigating the decidability of the theory of queues with integers and with subsequence relations including *subqueue*, *prefix* and *suffix* relations. This theory has a very strong expressive power; it can interpret the theory of word concatenation with length function, the theory of Presburger arithmetic with divisibility predicate, the theory of arrays, etc. Determining the decidability of this theory will have far-reaching consequences for solving other decision problems. In particular, it could give us a better understanding and classification of solutions to word equations. Besides the theoretical importance, the decidability of this theory may give us more powerful algorithms in pattern matching, which has numerous applications in computer science. It can precisely characterize the semantics of common string operations in the C language, and hence would be a powerful tool to reason about memory safety properties. It may also lead to a decision procedure for the theory of unbounded bit-vectors which potentially has many applications in hardware verification.

## References


1. Alessandro Armando, Silvio Ranise, and Michaël Rusinowitch. Uniform derivation of decision procedures by superposition. In *Proceedings of the 15th International Workshop on Computer Science Logic (CSL'01)* volume 2142 of *Lecture Notes in Computer Science*, pages 513–527. Springer-Verlag, 2001.
2. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, Cambridge, UK, 1999.
3. Rolf Backofen. A complete axiomatization of a theory with feature and arity constraints. *Journal of Logical Programming*, 24(1&2):37–71, 1995.
4. Michael Benedikt, Leonid Libkin, Thomas Schwentick, and Luc Segoufin. A model-theoretic approach to regular string relations. In *Proceedings of the 16th IEEE Symposium on Logic in Computer Science*, pages 431–440. IEEE Computer Society Press, 2001.

5. Nikolaj S. Bjørner. *Integrating Decision Procedures for Temporal Verification*. PhD thesis, Computer Science Department, Stanford University, November 1998.
6. Nikolaj S. Bjørner, Anca Browne, Michael Colón, Bernd Finkbeiner, Zohar Manna, Henny B. Sipma, and Tomás E. Uribe. Verifying temporal properties of reactive systems: A STeP tutorial. *Formal Methods in System Design*, 16(3):227–270, June 2000.
7. Hubert Comon. Solving symbolic ordering constraints. *International Journal of Foundations of Computer Science*, 1(4):387–411, 1990.
8. Hubert Comon and Catherine Delor. Equational formulae with membership constraints. *Information and Computation*, 112(2):167–216, 1994.
9. Hubert Comon and Pierre Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7:371–425, 1989.
10. Hubert Comon and Ralf Treinen. Ordering constraints on trees. In Sophie Tison, editor, *Proceedings of the 19th International Colloquium on Trees in Algebra and Programming (CAAP'94)*, volume 787 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 1994.
11. Hubert Comon and Ralf Treinen. The first-order theory of lexicographic path orderings is undecidable. *Theoretical Computer Science*, 176(1-2):67–87, 1997.
12. David C. Cooper. Theorem proving in arithmetic without multiplication. In *Machine Intelligence*, volume 7, pages 91–99. American Elsevier, 1972.
13. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2001.
14. Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 7:279–301, 1982.
15. Peter J. Downey, Ravi Sethi, and Robert E. Tarjan. Variations of the common subexpression problem. *Journal of ACM*, 27:758–771, 1980.
16. Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2001.
17. Solomon Feferman and Robert L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
18. Silvio Ghilardi. Model-theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning*, 33(3-4):221–249, 2005.
19. Wilfrid Hodges. *Model Theory*. Cambridge University Press, Cambridge, UK, 1993.
20. John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 1979.
21. Jean-Pierre Jouannaud and Mitsuhiro Okada. Satisfiability of systems of ordinal notation with the subterm property is decidable. In *Proceedings of the 18th International Colloquium on Automata, Languages and Programming*, volume 510 of *Lecture Notes in Computer Science*, pages 455–468. Springer-Verlag, 1991.
22. H. Jerome Keisler and Chen C. Chang. *Model Theory*. Elsevier Science, Netherlands, 1990.
23. Claude Kirchner, Hélène Kirchner, and Michaël Rusinowitch. Deduction with symbolic constraints. *Revue Francaise d' Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on automated deduction.
24. Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Proceedings of the 30th International Colloquium on Automata, Languages and Programming, ICALP'2003*, volume 2719 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.
25. Donald E. Knuth and Peter Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970. Reprinted in *Automation of Reasoning, Vol. 2* Jürgen Siekmann and G. Wrightson, editors, pages 342-376, Springer-Verlag, 1983.
26. Konstantin Korovin and Andrei Voronkov. A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering. In *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, pages 291 – 302. IEEE Computer Society Press, 2000.

27. Konstantin Korovin and Andrei Voronkov. Knuth-Bendix constraint solving is NP-complete. In *Proceedings of 28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 979–992. Springer-Verlag, 2001.
28. Konstantin Korovin and Andrei Voronkov. The decidability of the first-order theory of the Knuth-Bendix order in the case of unary signatures. In *Proceedings of the 22th Conference on Foundations of Software Technology and Theoretical Computer Science, (FSTTCS'02)*, volume 2556 of *Lecture Notes in Computer Science*, pages 230–240. Springer-Verlag, 2002.
29. Viktor Kuncak and Martin Rinard. On the theory of structural subtyping. Technical Report MIT-LCS-TR-879, Massachusetts Institute of Technology, January 2003.
30. Viktor Kuncak and Martin Rinard. The structural subtyping of non-recursive types is decidable. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science*, pages 96–107. IEEE Computer Society Press, 2003.
31. Viktor Kuncak and Martin Rinard. An algorithm for deciding BAPA: Boolean algebra with Presburger arithmetic. In *Proceedings of the 20th International Conference on Automated Deduction (CADE'05)*, volume 3632 of *Lecture Notes in Computer Science*, pages 260–277. Springer-Verlag, 2005.
32. M. Lothaire. *Combinatorics on Words*. Addison-Wesley, Massachusetts, USA, 1983. M. Lothaire is a joint pseudonym for the following: Robert Cor, Dominque Perrin, Jean Berstel, Christian Choffrut, Dominque Foata, Jean Eric Pin, Guiseppe Pirillo, Christophe Reutenauer, Marcel P. Schutzenberger, Jadcques Sakaroovitch, and Imre Simon.
33. László Lovász. *Combinatorial Problems and Exercises*. Elsevier, Horth-Holland, 1993.
34. Michael J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite tree. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, pages 348–357. IEEE Computer Society Press, 1988.
35. Gennady S. Makanin. The problem of solvability of equations in a free semigroup. *Math. Sbornik*, 103:147–236, 1977.
36. Anatoly I. Malcev. Axiomatizable classes of locally free algebras of various types. In *The Metamathematics of Algebraic Systems, Collected Papers*, chapter 23, pages 262–281. North Holland, 1971.
37. Paliath Narendran and Michaël Rusinowitch. The theory of total unary RPO is decidable. In *Proceedings of the 1st International Conference on Computational Logic (CL 2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 660–672. Springer-Verlag, 2000.
38. Paliath Narendran, Michaël Rusinowitch, and Rakesh M. Verma. RPO constraint solving is in NP. In *Proceedings of the 12th International Workshop on Computer Science Logic (CSL 98)*, volume 1584 of *Lecture Notes in Computer Science*, pages 385 – 398. Springer-Verlag, 1999.
39. Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transaction on Programming Languages and Systems*, 1(2):245–257, October 1979.
40. Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure. *Journal of ACM*, 27(2):356–364, April 1980.
41. Robert Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letters*, 47(2):65–69, 1993.
42. Robert Nieuwenhuis and José M. Rivero. Solved forms for path ordering constraints. In *Proceeding of 10th International Conference on Rewriting Techniques and Applications (RTA)*, volume 1631 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 1999.
43. Robert Nieuwenhuis and Albert Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19(4):321–351, 1995.
44. Derek C. Oppen. Elementary bounds for Presburger arithmetic. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing*, pages 34–37. ACM Press, 1973.
45. Derek C. Oppen. Reasoning about recursively defined data structures. *Journal of ACM*, 27(3), July 1980.

46. Cattamanchi R. Reddy and Donald W. Loveland. Presburger arithmetic with bounded quantifier alternation. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 320–325. ACM Press, 1978.
47. Peter Revesz. Quantifier-elimination for the first-order theory of boolean algebras with linear cardinality constraints. In *Proceedings of the 18th Conference on Advances in Databases and Information Systems (ADBIS'04)*, volume 3255 of *Lecture Notes in Computer Science*, pages 1–21. Springer-Verlag, 2004.
48. Tatiana Rybina and Andrei Voronkov. A decision procedure for term algebras with queues. In *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, pages 279 – 290. IEEE Computer Society Press, 2000.
49. Tatiana Rybina and Andrei Voronkov. A decision procedure for term algebras with queues. *ACM Transactions on Computational Logic*, 2(2):155–181, 2001.
50. Tatiana Rybina and Andrei Voronkov. Upper bounds for a theory of queues. In *Proceedings of 30th International Colloquium on Automata, Languages and Programming (ICALP'03)*, volume 2719 of *Lecture Notes in Computer Science*, pages 714–724. Springer-Verlag, 2003.
51. Thoralf A. Skolem. Untersuchungen über die Axiome des Klassenkalküls und über Produktations- und Summationsprobleme, welche gewisse Klassen von Aussagen betreffen. In Jens Erik Fenstad, editor, *Selected works in logic*, pages 67–101. Universitetsforlaget, 1970.
52. Wolfgang Thomas. Infinite trees and automaton-definable relations over $\omega$-words. *Theoretical Computer Science*, 103:143–159, 1992.
53. Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In F. Baader and Klaus U. Schulz, editors, *Proceedings of the 1st International Workshop on Frontiers of Combining Systems (FroCos'96)*, Applied Logic Series, Vol. 3, pages 103–120. Kluwer Academic Publishers, 1996.
54. Cesare Tinelli and Christophe Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 290(1):291–353, January 2003.
55. Cesare Tinelli and Calogero G. Zarba. Combining decision procedures for sorted theories. In José Júlio Alferes and João Alexandre Leite, editors, *Proceedings of the 9th European Conference on Logic in Artificial Intelligence (JELIA'04)*, volume 3229 of *Lecture Notes in Computer Science*, pages 641–653. Springer-Verlag, 2004.
56. Ralf Treinen. A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation*, 14:437–457, 1992.
57. K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *Journal of ACM*, 34(2):492–510, 1987.
58. Calogero G. Zarba. Combining multisets with integers. In Andrei Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 363–376. Springer-Verlag, 2002.
59. Calogero G. Zarba. Combining sets with integers. In Alessandro Armando, editor, *Proceedings of the 4th International Workshop on Frontiers of Combining Systems (FroCoS'02)*, volume 2309 of *Lecture Notes in Artificial Intelligence*, pages 103–116. Springer-Verlag, 2002.
60. Ting Zhang, Henny B. Sipma, and Zohar Manna. Decision procedures for recursive data structures with integer constraints. In *Proceedings of the 2nd International Joint Conference on Automated Reasoning (IJCAR'04)*, volume 3097 of *Lecture Notes in Computer Science*, pages 152–167. Springer-Verlag, 2004.
61. Ting Zhang, Henny B. Sipma, and Zohar Manna. Term algebras with length function and bounded quantifier alternation. In *Proceedings of the 17th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'04)*, volume 3223 of *Lecture Notes in Computer Science*, pages 321–336. Springer-Verlag, 2004.
62. Ting Zhang, Henny B. Sipma, and Zohar Manna. The decidability of the first-order theory of term algebras with Knuth-Bendix order. In Robert Nieuwenhuis, editor, *the* $20^{th}$ *International Conference on Automated Deduction (CADE'05)*, volume 3632 of *Lecture Notes in Computer Science*, pages 131–148. Springer-Verlag, 2005.

63. Ting Zhang, Henny B. Sipma, and Zohar Manna. Decision procedures for queues with integer constraints. In R. Ramanujam and Sandeep Sen, editors, *Proceedings of the 25th International Conference on the Foundatations of Software Technology and Theoretical Computer Science (FSTTCS'05)*, volume 3821 of *Lecture Notes in Computer Science*, pages 225–237. Springer-Verlag, 2005.
64. Ting Zhang, Henny B. Sipma, and Zohar Manna. Decision procedures for term algebras with integer constraints. *Information and Computation*, 204(10):1526–1574, October 2006.
65. Zohar Manna, Henny B. Sipma, and Ting Zhang. Verifying Balanced Trees. *Symposium on Logical Foundations of Computer Science*, 2007. To appear.