Decision Procedures for Proving Inductive Theorems without Induction
Takahito Aoto, Sorin Stratulat

To cite this version:
Takahito Aoto, Sorin Stratulat. Decision Procedures for Proving Inductive Theorems without Induction. 16th International Symposium on Principles and Practice of Declarative Programming (PPDP) 2014, Sep 2014, Canterbury, United Kingdom. 10.1145/2643135.2643156. hal-01098929

HAL Id: hal-01098929
https://hal.science/hal-01098929
Submitted on 7 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract
Automated inductive reasoning for term rewriting has been extensively studied in the literature. Classes of equations and term rewriting systems (TRSs) with decidable inductive validity have been identified and used to automatize the inductive reasoning. We give procedures for deciding the inductive validity of equations in some standard TRSs on natural numbers and lists. Contrary to previous decidability results, our procedures can automatically decide without involving induction reasoning the inductive validity of arbitrary equations for these TRSs, that is, without imposing any syntactical restrictions on the form of equations. We also report on the complexity of our decision procedures. These decision procedures are implemented in our automated provers for inductive theorems of TRSs and experiments are reported.

Categories and Subject Descriptors F.3.1 [Specifying and Verifying and Reasoning about Programs]: Mechanical verification; F.4.1 [Mathematical Logic]: Mechanical theorem proving; I.2.3 [Deduction and Theorem Proving]: Deduction; F.4.2 [Grammars and Other Rewriting Systems]: Decision problems

Keywords Inductive Theorems, Term Rewriting Systems, Decision Procedure, Initial Algebra

1. Introduction
Inductive reasoning on recursively defined data structures is ubiquitous in the verification of formal specifications and software. In equational logic, the properties to be checked are formalized as inductive theorems of term rewriting systems (TRSs for short). It is known that the methods for automatically proving inductive theorems of TRSs easily diverge, the construction of effective inductive theorem provers still remaining a hard challenge [15].

In [21], Kapur and Subramaniam initiated the problem of identifying classes of conjectures and TRSs for which automated inductive proof methods provide decision procedures. More precisely, they gave syntactic conditions on structure of (recursive) function definitions and of conjectures, and showed that if one runs a (prefixed) implicit induction method in a (prefixed) particular strategy for any conjecture and TRS satisfying these conditions, then it never diverges and positive/negative answer is found always. This approach has been extended with other authors to more general TRSs and classes of conjectures in [11–14, 23]. Our work is motivated by strengthening the power of automated inductive reasoning, by invoking such decision procedures for inductive theorems, inside automated induction provers such as the authors’ [2] [28]. It is well accepted that, often, a key ingredient of successful induction reasoning is the use of subsidiary lemmas [13], and thus various methods for automatically generating lemmas have been inspected [3, 26, 41, 52]. But, as one might expect, lemma generation methods often generate many incorrect conjectures. Even if decision procedures are only effective for restricted subclasses, and even if the given conjecture and TRS do not fall inside the scope of these classes, decision procedures could be helpful for solving these lemma candidates, often automatically generated while searching a successful inductive reasoning.

The decidability results obtained by the approach mentioned above, however, turned out to be not very helpful for this purpose. This is because usually conjectures satisfying the syntactic conditions of decision procedures can already be proved solely by the automated induction provers, as these decision procedures and the automated induction provers are basically based on similar induction methods. This motivates us to investigate different approaches for obtaining classes of equations and TRSs with decidable inductive validity.

In this paper, we propose a new approach for deciding inductive theorems of TRSs. Our essential idea is to use the validity in the initial algebras of TRSs, instead of the validity guaranteed by the existence of inductive proofs. For equations and TRSs, it is known that the inductive validity and the validity in initial algebras coincide. Thus, if we can decide the validity in initial algebras of TRSs, then we get a decision procedure for inductive theorems of TRSs. Furthermore, this approach is completely different from finding induction proofs and it does not suffer from the weakness of the Kapur and Subramaniam’s approach when used inside the general automated induction proving methods. Our approach seems very natural but, at the best of our knowledge, such approach for proving inductive theorems has not been investigated, albeit the usability of decidable arithmetic theories for building induction schemes has been investigated in [8] [20]. An obvious weakness of our approach is that it works only for specific TRSs. On the other hand, our decision procedures impose no syntactical conditions on the equations and do not require induction reasoning, contrary to the known decision procedures for inductive theorems. It may be also considered as a weakness that our approach does not provide induction proofs, which may be helpful to give a strategy for constructing proofs in formal proof systems such as Isabelle/HOL [24]. On the other hand, our approach may have a similarity to the normalization by evaluation technique, which
accomplishes the syntactic goal (normalization) using semantics (evaluation).

Because of the nature of our approach, our decision procedures consist in checking the validity of equations in some models of TRSs, i.e., in algebras. Hence, our decision procedures presented can be fallen in the more general category of automated proving methods. Naturally, the decidability of Presburger arithmetic (PA) turned out to be very useful. In fact, one of our decidability results is subsumed by the one using encoding to PA formulas theoretically. Throughout the paper, we also explain when known methods for proving the validity of (initial) algebras are available—but we would stress here that our central idea is rather not the introduction of the decision procedures for these algebras, but their application to obtain decision procedures for inductive theorems.

The rest of the paper is organized as follows. Section 2 covers preliminaries. In Section 3, we give an exponential procedure for deciding the inductive validity of the TRS consisting of the addition and multiplication on natural numbers. Then, we extend our decision procedure by incorporating some standard list functions such as append, reverse and length in Section 4. In Section 5, we present some decision procedures for the inductive theorems of the TRS with max and min functions on natural numbers. In Section 6, we report on the implementation and experiments. Section 7 concludes. Some of proofs have been put in the appendix.

2. Preliminaries

We assume basic familiarity with term rewriting and semantics of equational logic [9].

A many-sorted signature \( \Sigma = \langle S, F \rangle \) consists of the set \( S \) of sorts and the set \( F \) of many-sorted function symbols; \( f : \alpha_1 \times \cdots \times \alpha_n \rightarrow \alpha_0 \), with \( \alpha_i \in S \) and \( n \geq 0 \), denotes the signature of a function symbol \( f \in F \). If \( S \) is a singleton set, say \( \{\alpha\} \), then the many-sorted signature is called a first-order signature. In this case, \( \alpha_1 \times \cdots \times \alpha_n \rightarrow \alpha_0 \) is abbreviated by \( n \)-ary, and \( f : \alpha_1 \times \cdots \times \alpha_n \rightarrow \alpha_0 \) is denoted by \( f^n \).

The \( S \)-sorted variables (or variables) are \( V = \bigcup_{\alpha \in S} V^\alpha \), where each \( V^\alpha \) is disjoint from the others. The set \( T(S, V^\alpha) \) of \( \alpha \)-terms consists of the terms \( \prod (t(\Sigma, V)^\alpha) \) of terms (or terms) of sort \( \alpha \in S \) inductively defined by \( (1) \quad V(\alpha) \subseteq T(S, V)^\alpha \) and \( (1) \quad f \in F, \quad \alpha_1 \times \cdots \times \alpha_n \rightarrow \alpha \in F \) and \( t_1, \ldots, t_n \in T(S, V)^\alpha \). Then, \( f(t_1, \ldots, t_n) \in T(S, V)^\alpha \) for \( i = 1, \ldots, n \), and \( f(t_1, \ldots, t_n) \in T(S, V)^\alpha \) for \( i = 1, \ldots, n \), and \( t \in T(S, V)^\alpha \).

The set of terms is given by \( T(\Sigma, V) = \bigcup_{\alpha \in S} T(S, V)^\alpha \). The set of variables (function symbols) in a term \( t \) is denoted by \( V(t) \) (\( F(t) \), respectively). A term \( \alpha \) is said to be ground if \( V(t) = \emptyset \). We denote an empty sequence by \( \varepsilon \), and call it an ground instance of \( \Sigma \).

We denote an empty sequence by \( \varepsilon \), and call it an ground instance of \( \Sigma \). A many-sorted signature \( \Sigma = \langle S, F \rangle \), a \( \Sigma \)-algebra \( A = \langle \alpha \rangle_{\alpha \in S}, \langle f^A \rangle_{f \in F} \rangle \) is a pair of tuples where \( \alpha \in S \) are mutually disjoint, and \( f^A \) is a mapping \( A^\alpha \times \cdots \times A^n \rightarrow A^\alpha \), for each \( f : \alpha_1 \times \cdots \times \alpha_n \rightarrow \alpha \in F \). The set \( A = \bigcup_{\alpha \in A} A^\alpha \) is called the carrier set of the \( \Sigma \)-algebra \( A \) and denoted by \( |A| \). If \( S = \{\alpha_1, \ldots, \alpha_n\} \) and \( F = \{f_1, \ldots, f_k\} \), \( A \) is written like \( \langle A^\alpha_1, \ldots, A^\alpha_n; f_1^A, \ldots, f_k^A \rangle \). The \( \Sigma \)-term algebra \( A = \langle \alpha \rangle_{\alpha \in S}, \langle f^A \rangle_{f \in F} \rangle \) given by \( A^\alpha \) for each \( \alpha \in S \) and \( f^A(s_1, \ldots, s_n) = f(s_1, \ldots, s_n) \) for each \( f \in F \). The \( \Sigma \)-term algebra is denoted by \( T(\Sigma, V) \).

We define a ground \( \Sigma \)-term algebra as \( T_{\Sigma}(V) \), which will be denoted by \( T_{\Sigma}(V) \).

A valuation \( \rho \) on a \( \Sigma \)-algebra \( A \) is a tuple \( \rho = \rho_{\alpha} \in A^\alpha \) of mappings \( \rho_{\alpha} : V^\alpha \rightarrow A^\alpha \). We abbreviate \( \rho_{\alpha}(x) \) (with \( x \in V^\alpha \)) as \( \rho(x) \).

Given a many-sorted signature \( \Sigma = \langle S, F \rangle \) and a \( \Sigma \)-algebra \( A = \langle \alpha \rangle_{\alpha \in S}, \langle f^A \rangle_{f \in F} \rangle \), we define the interpretation \( \llbracket t \rrbracket_A \) (which is abbreviated as \( \llbracket t \rrbracket \) for brevity) of a \( \alpha \)-term \( t \) on \( A \) w.r.t. a valuation \( \rho \) on \( A \) like this: \( \llbracket t \rrbracket = \rho(x) \) and \( \llbracket f(t_1, \ldots, t_n) \rrbracket = f^A(\llbracket t_1 \rrbracket, \ldots, \llbracket t_n \rrbracket) \). It is easily shown that \( \llbracket t \rrbracket \in A^\alpha \), for each \( t \in T(S, V)^\alpha \). A \( \alpha \)-equation \( s \equiv t \in \Sigma \) is valid on a \( \Sigma \)-algebra \( A \) (denoted by \( A \models s \equiv t \)) if \( \llbracket s \rrbracket = \llbracket t \rrbracket \), for any valuation \( \rho \) on \( A \).

For set equality of equations, \( A \models E \) is defined by \( A \models s \equiv t \) for all \( s, t \in E \).

Let \( A = \langle \alpha \rangle_{\alpha \in S}, \langle f^A \rangle_{f \in F} \rangle, B = \langle \beta \rangle_{\beta \in S}, \langle f^B \rangle_{f \in F} \rangle \) be \( \Sigma \)-algebras. A \( \Sigma \)-homomorphism \( \sigma \) from \( A \) to \( B \) is a tuple of mappings \( \sigma_{\alpha} : A^\alpha \rightarrow B^\alpha \) such that \( \sigma_{\alpha}(f^A(a_1, \ldots, a_n)) = f^B(\sigma_{\alpha}(a_1), \ldots, \sigma_{\alpha}(a_n)) \), for each \( f : \alpha_1 \times \cdots \times \alpha_n \rightarrow \beta \in F \). If \( S = \{\alpha\} \) then \( \sigma \) is identified with \( \sigma_{\alpha} \). Two \( \Sigma \)-algebras \( A \) and \( B \) are isomorphic (denoted by \( A \cong B \)) if there exists a \( \Sigma \)-homomorphism \( \sigma \) consisting of bijective mappings. Isomorphic \( \Sigma \)-algebras can be often identified.

Let \( \Sigma = \langle S, F \rangle \) be a signature and \( A = \langle \alpha \rangle_{\alpha \in S}, \langle f^A \rangle_{f \in F} \rangle \) a \( \Sigma \)-algebra. An equivalence relation \( \sim \) on \( A \) is said to be a congruence (on \( A \)) if \( (a \sim b) \) implies \( f(a_1, \ldots, a_n) \sim f(b_1, \ldots, b_n) \). We denote the \( \sim \)-equivalence class of \( a \in A \) by \( [a]_{\sim} \). If \( \sim \) is a congruence on \( A \) then we obtain its quotient \( \Sigma \)-algebra \( A/\sim = \langle \langle A \rangle_{\sim}/\alpha \rangle_{\alpha \in S}, \langle f_{\sim}/\alpha \rangle_{f \in F} \rangle \) by defining \( (A/\sim)^\alpha = \{[a]_{\sim} \mid a \in A^\alpha \} \) and \( (A/\sim)^f = \langle [a_1]_{\sim}, \ldots, [a_n]_{\sim} \rangle = f_{\sim}([a_1]_{\sim}, \ldots, [a_n]_{\sim}) \).

Let \( K \) be a class of \( \Sigma \)-algebras. A \( \Sigma \)-algebra \( A \) is said to be initial in \( K \) if, for any \( B \in K \), there exists a unique \( \Sigma \)-homomorphism \( A \rightarrow B \). The initial algebras are unique up to isomorphism. Let \( R \) be a TRS, and \( K \) be the class of \( \Sigma \)-algebras satisfying \( R \), i.e., \( K = \{A \mid A \models R \} \). Then the quotient ground \( \Sigma \)-term algebra \( T_{\Sigma} \) is initial in \( K \), which is called the initial \( \Sigma \)-algebra of \( R \).
and denoted by \( \mathcal{I}_R \). Validity on initial algebras and inductive theorems correspond in the following way:

**Proposition 2.1** (e.g., [19]). Let \( \mathcal{R} \) be a TRS over signature \( \Sigma \) and \( s, t \in T(\Sigma, \mathcal{V}) \). An equation \( s \equiv t \) is an inductive theorem of \( \mathcal{R} \) iff it is valid in the initial \( \Sigma \)-algebra of \( \mathcal{R} \).

### 3. Deciding Inductive Theorems on Natural Numbers with Addition and Multiplication

In this section, we consider a first-order signature \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) = \langle \{ N \}, \mathcal{F}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \rangle \), where \( \mathcal{F}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) = \{ (\times(\cdot, \cdot), \cdot(\cdot, \cdot)) \} \).

Let \( \mathcal{R}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \) be the following TRS over \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \) that defines the multiplication and addition on natural numbers encoded by 0 and the successor function \( + \).

\[
\mathcal{R}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) = \begin{cases}
+0(y) & \rightarrow y \\
+(s(x), y) & \rightarrow s((x, y)) \\
(x, y) & \rightarrow 0 \\
(s(x), y) & \rightarrow +(x, y, y)
\end{cases}
\]

We present a decision procedure for \( \mathcal{R}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \) in the following way:

**Lemma 3.1.** The initial \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \)-algebra of \( \mathcal{R}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \) is isomorphic to \( \mathcal{N} \times \mathcal{N} \).

Let us next consider a first-order signature \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) = \langle \{ N \}, \mathcal{F}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \rangle \), where \( \mathcal{F}(\mathbb{N} \times \mathbb{N}, \mathcal{V}) = \{ (\times(\cdot, \cdot), \cdot(\cdot, \cdot)) \} \).

The sets of natural numbers and integers equipped with the usual operations of multiplication, addition, 1 and 0 form \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \)-algebras \( \mathcal{N} \times \mathcal{N} \) and \( \mathbb{Z} \), which will be abbreviated as \( \mathbb{N} \) and \( \mathbb{Z} \), respectively, in what follows.

For \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \)-algebra \( \mathcal{K} = \langle \langle K; \times, +, 1, 0 \rangle \rangle \), the carrier set \( K \) will be identified with \( \mathcal{K} \). Let \( \Sigma(\mathbb{N} \times \mathbb{N}, \mathcal{V}) \)-algebra \( \mathcal{K} = \langle \langle K; \times, +, 1, 0 \rangle \rangle \) be commutative ring. A zero ring is a ring with a singleton carrier set where we have \( 0 = 1 \). A nonzero commutative ring is said to be an integral domain if \( a \times b \neq 0 \) for any \( a, b \in K \) such that \( a \neq 0 \) and \( b \neq 0 \).

The polynomial ring over a commutative ring \( \mathcal{K} \) and indeterminates \( x_1, \ldots, x_n \) is denoted by \( \mathcal{K}[x_1, \ldots, x_n] \). Elements of \( \mathcal{K}[x_1, \ldots, x_n] \) are called \( \mathcal{K} \)-polynomials; each element has its \( \mathcal{K} \)-algebra, and \( \mathcal{K} \)-polynomials are encoded by 0 and the successor function \( + \).

\[
\mathcal{K}[x_1, \ldots, x_n] = \mathcal{K} \times \mathcal{K} \cup \mathcal{K}[x_1, \ldots, x_{n-1}, x_n]
\]

The complexity of derivations in an equational proof system for formal polynomials has been studied, e.g., in [19, 25]. Thus, randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18]. Randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18]. Randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18]. Randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18]. Randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18]. Randomized algorithms for effectively checking the identity of two formal polynomials have been studied, e.g., in [18].
Let $\Sigma_{(+,0)} = \{(N), F_{(+,0)}\}$, where $F_{(+,0)} = \{ (+^{(2)}, s^{(1)}), 0^{(0)} \}$, and $R_{(+)} = \{ (r \rightarrow r \in R_{(x,+)} ) | l(e) = + \}$. It is also easy to see the following.

**Theorem 3.7.** It is decidable in polynomial time whether for given $s, t \in T(\Sigma_{(+,0)}, \mathcal{V})$ the equation $s \approx t$ is an inductive theorem of $R_{(+)}$.

From the perspective of deciding the validity on natural numbers, the above results let us think that the situation is much nicer for equations than for general first-order formulas: for an arbitrary first-order formula $\varphi$ over $\Sigma_{(+,0)}$, $R_{(+,0)}$ is $\varphi$ (Peano arithmetic) is undecidable, and even for the one over the signature $\Sigma_{(+,0)}$, the complexity of deciding $N_{(+,0)} \models \varphi$ (PA) is doubly exponential (e.g. (17)).

In fact, an exponential-time decision procedure for checking the validity of universal PA formulas is known (27), and one can use this to decide $N_{(+,0)} \models s \approx t$.

Similar to the Peano arithmetic case, one may be tempted to expect deciding the validity of equations on natural numbers is often decidable—but it is not true: in fact, extending $N_{(+,0)}$ with a simple function is enough to make the inductive validity of equations undecidable. This follows from the well-known result on the Hilbert 10th Problem (22): it is undecidable whether $\exists a_1, \ldots, a_n \in \mathbb{Z}$, $\varphi(a_1, \ldots, a_n) = 0$ for a given $\varphi \in \mathcal{L}[x_1, \ldots, x_n]$. We now explain this, as the undecidability of inductive theorems seems to be folklore which we could not find in the literature.

Let us consider a first-order signature $\Sigma_{(eq, +, 0)} = \{(N), \{eq^{(2)}\} \cup F_{(+,0)}\}$. Let $R_{(eq,x,+)}$ be the followingTRS over $\Sigma_{(eq,x,+)}$.

$$
R_{(eq,x,+)} = R_{(x,+)} \cup \left\{ \begin{array}{c}
eq(0,0) \rightarrow s(0) \\
\neq(s(x),0) \rightarrow 0 \\
\neq(0,s(x)) \rightarrow 0 \\
\neq(s(x),s(y)) \rightarrow \neq(x,y) \end{array} \right. $$

In the proof of Theorem 3.5 we showed that the equality of two formal polynomials ($s[l]$) and ($l[t]$) is decidable, for all natural numbers (i.e. $(l[s]) = (l[t])$). The idea here is to use the fact that it is undecidable whether $(l[s])$ (and $(l[t])$) are different for all natural numbers. By the additional eq-rules, this problem can be encoded as whether $\neq(s,t) \approx 0$ is an inductive theorem.

**Proposition 3.8.** (Toyama (20)). It is undecidable whether for given $s, t \in T(\Sigma_{(eq,x,+),0}, \mathcal{V})$ the equation $s \approx t$ is an inductive theorem of $R_{(eq,x,+)}$.

### 4. Deciding Inductive Theorems on Lists of Natural Numbers

In this section, we extend the decidability results of previous sections to lists of natural numbers. For this, we assume a set $F_{(len, rev, 0, x, +, \ldots, nil, s, 0)} = F_{(+, s, 0)} \cup \{ len^{L-N}, rev^{L-L}, L x L \rightarrow L x L, :x, :y \}$ of function symbols and consider the many-sorted signature $\Sigma_{(len, rev, 0, x, +, \ldots, nil, s, 0)} = \langle \{N, L\}, F_{(len, rev, 0, x, +, \ldots, nil, s, 0)} \rangle$. We define the following TRS $R_{(len, rev, 0, x, +, \ldots, nil, s, 0)}$ over $\Sigma_{(len, rev, 0, x, +, \ldots, nil, s, 0)}$ that encodes append, reverse and length functions on lists of natural numbers, where nil and :: are the list constructors: $R_{(len, rev, 0, x, +, \ldots, nil, s, 0)} = \left\{ \begin{array}{c}
\neq(\text{nil}, y) \rightarrow y \\
\neq((x,x), y) \rightarrow (x, \neq((x,y), y)) \\
\text{rev}(\text{nil}) \rightarrow \text{nil} \\
\text{rev}((x,x)) \rightarrow \neq(\text{rev}(x), x) \\
\text{len}(\text{nil}) \rightarrow 0 \\
\text{len}(\text{rev}(x)) \rightarrow s(\text{len}(x)) \\
\end{array} \right. $

We are now going to present a procedure to decide whether $R_{(len, rev, 0, x, +, \ldots, nil, s, 0)}$ is consistent for given terms $s, t \in T(\Sigma_{(len, rev, 0, x, +, \ldots, nil, s, 0)}, \mathcal{V})$. In the following, we will often abbreviate the subscripts (len, rev, 0, x, +, ::, nil, s, 0) and (len, rev, 0, x, +, ::) as (len, ...).

Let $\Sigma_{(nil, s, 0)} = \{(N, L), \{(\text{nil}^{L-N}, \text{nil}^{L-N} + N, 0^{N})\}$ be a many-sorted signature. Consider a $\Sigma_{(len, nil, s, 0)}$-algebra $R_{(len, nil, s, 0)}$ of $\Sigma_{(len, rev, 0, x, +, \ldots, nil, s, 0)}$-algebra. In the rest of this subsection, $L_{(\text{len}, \text{nil}, \text{s}, 0)}$ is abbreviated as $L$. In our decision procedure, we consider normal form defined in terms of the list symbol single instead of ::. The term single($x$) is interpreted as the singleton list ::(x, nil). Thus we consider a signature $\Sigma_{(\text{single}, \text{len}, \text{rev}, 0, x, +, \ldots, \text{nil}, s, 0)} = \langle \{N, L\}, \{\text{single}^{L-N}, \text{rev}^{L-L}, \text{len}^{L-N}\} \rangle$. We abbreviate $\{\text{single}^{L-N}, \text{rev}^{L-L}, \text{len}^{L-N}\}$ as $F_{(\text{single}, \ldots)}$ and $\Sigma_{(\text{single}, \text{len}, \text{rev}, 0, x, +, \ldots, \text{nil}, s, 0)}$ as $\Sigma_{(\text{single}, \ldots)}$.

**Procedure list-check(s,t)**

1. Normalize $s$ and $t$ by the following TRS $S_{(\text{len}, \text{rev}, \ldots)} =$

$$
\begin{array}{c}
\neq(x, \text{nil}) \rightarrow x \\
\neq(\text{nil}, y) \rightarrow y \\
\neq((x,x), y) \rightarrow \neq((x,y), y) \\
\text{rev}(\text{nil}) \rightarrow \text{nil} \\
\text{rev}(\text{single}(x)) \rightarrow \text{single}(x) \\
\text{rev}(\text{rev}(x)) \rightarrow x \\
\text{len}(\text{nil}) \rightarrow 0 \\
\text{len}(\text{single}(x)) \rightarrow s(0) \\
\text{len}(\text{rev}(x)) \rightarrow \text{len}(x) \\
\text{len}(0, \text{rev}(x)) \rightarrow + (\text{len}(x), \text{len}(y)) \\
\end{array}
$$

Note that $S_{(\text{len}, \text{rev}, \ldots)}$ is convergent and each term has a unique $S_{(\text{len}, \text{rev}, \ldots)}$-normal form. In the rest of this section, we will abbreviate $S_{(\text{len}, \text{rev}, \ldots)}$ as $S$, and the $S_{(\text{len}, \text{rev}, \ldots)}$-normal form of a term $u$ as $u_+$.

As we will see, if len($u$) is a subterm of a $\Sigma_{(\text{len}, \ldots)}$-normal term of sort $N$ then $u \in V$. To deal with such terms, let $\text{len}^V = \{(\text{len}(x) | x \in V)\}$, and $\Sigma_{(\text{len}, \text{rev}, \text{rev}, \ldots)} = \langle \{N, L\}, F_{(\text{rev}, \text{rev}, \ldots)} \rangle$.

2. Define (elem)-terms and (list)-terms by the following (extended) BNF.

**BNF**

(elem)-terms $v_1 ::= x s | \text{single}(x) | \text{rev}(x)$

(list)-terms $v_1 ::= \text{nil} | v_1 @ v_2 @ \ldots @ v_k$
where \(xs\) ranges over \(V_L\), and \(u\) ranges over \(T(\Sigma(x,+,\cdot,\min,0), V)^N\). Here, \(\oplus\) is assumed to be associative. As we will see, \(s_1, t_1\) are either terms in \(T(\Sigma(x,+,\cdot,\min,0), V)^N\) (if they have sort \(N\)) or (list-)terms (if they have sort \(L\)).

3. Consider a set \(V^N\) of new variables that is bijective to \(\text{len} (V^L)\). Let \(\delta: \text{len} (V^L) \to V^N\) be a bijection. Furthermore, for any \(u \in T(\Sigma(x,+,\cdot,\min,0), V)^N\), let \(\text{Abs}(u)\) be the term in \(T(\Sigma(x,+,\cdot,\min,0), V^N \cup \text{len} (V^N))\) obtained by replacing every \(\text{len}(x) \in \text{len} (V^L)\) with \(\delta(\text{len}(x))\).

   If \(s_1, t_1\) have sort \(N\), then check \(\{\Delta(\text{Abs}(s_1))\} = \{\Delta(\text{Abs}(t_1))\}\), i.e., whether \(\text{Abs}(s_1)\) and \(\text{Abs}(t_1)\) are the same formal polynomial, and use Theorem 3.5.

4. Consider now the remaining case when \(s_1, t_1\) have the sort \(L\).

   For any (list-)term \(w\), let \(\text{List}(w)\) be the list of its (elem-)terms, i.e., if \(w = \text{nil}\) then \(\text{List}(w) = []\) and if \(w = v_1 \oplus \cdots \oplus v_k\) then \(\text{List}(w) = [v_1, \ldots, v_k]\).

   Take \(l_1 = \text{List}(s_1)\) and \(l_2 = \text{List}(t_1)\). Then, check \(l_1\) and \(l_2\) have equal lengths, and that, for each pair \((v_i, v_j)\) of \(i\)-the elements of \(l_1\) and \(l_2\), either \(v_i = v_j \in V^L\), or \(v_i = \text{rev}(x) = v'_i\) for some \(xs \in V^L\), or \(v_i = \text{single} (u)\), \(v'_i = \text{single} (u)\) for some \(u, u'\), \(\text{Abs}(u) = \text{Abs}(u')\), holds.

**Lemma 4.2.** Let \(s \in T(\Sigma(\text{single},\cdot), V)\). If \(s\) has sort \(N\) then \(\text{Abs}(s)\) can be computed in polynomial time. (2) If \(s\) has sort \(L\) then \(\text{List}(s)\) can be computed in polynomial time.

It is easy to show that \(\Rightarrow \cup \Rightarrow\) is terminating, where \(\Rightarrow\) is the proper subterm relation. This fact is a basis of the proof of the following Lemma.

**Lemma 4.3.** Let \(s \in T(\Sigma(\text{single},\cdot), V)\). If \(s\) has sort \(N\) then \(s\) is a (list-)term.

Note in the next lemma that it follows from Lemma 4.3 and \(T(\Sigma(\text{single},\cdot), V)^N \subseteq T(\Sigma(\text{single},\cdot), V)^N\) that (1) for any \(s \in T(\Sigma(\text{single},\cdot), V)^N\), \(s\) is a (list-)term, and hence \(\text{Abs}(s)\) is defined, and (2) for any \(s \in T(\Sigma(\text{single},\cdot), V)^N\), \(s\) is a (list-)term and thus \(\text{List}(s)\) is defined.

**Lemma 4.4.** 1. For \(s, t \in T(\Sigma_{\langle n \rangle}, V)^N\), \(L \vdash s \equiv t\) if 
\[ N \models = \text{Abs}(s) = \text{Abs}(t). \]
2. Let \(s, t \in T(\Sigma_{\langle n \rangle}, V)^L\), \(\text{List}(s) = [u_1, \ldots, u_k]\) and \(\text{List}(t) = [v_1, \ldots, v_l]\). Then \(L \vdash s \equiv t\) if \(k = l\) and for every \(i = 1, \ldots, k,\) either (i) \(u_i = v_i \in V^L\), (ii) \(u_i = \text{rev}(x) = v_i\) for some \(xs \in V^L\), or (iii) \(u_i = \text{single} (u)\) and \(v_i = \text{single} (v)\) with \(N \models = \text{Abs}(u) = \text{Abs}(v)\).

Now we arrive at the main theorem of this section, claiming the decidability of a certain validity of \(\mathcal{R}(\text{rev}, \text{rev}, \odot, +, x)\).

**Theorem 4.5.** It is decidable in exponential time for given \(s, t \in T(\Sigma_{\langle n \rangle}, V)^L\), \(s \equiv t\) is an inductive theorem of \(\mathcal{R}(\text{rev}, \text{rev}, \odot, +, x)\).

**Proof:** We first claim \(\text{list-check}(s, t)\) can be done in exponential time. If \(s, t\) have sort \(N\), then, by Lemma 4.2, \(\text{Abs}(s), \text{Abs}(t)\) can be computed in polynomial time. Then, \(\text{Abs}(s)\) is a (list-)term, and \(\text{Abs}(t)\) can be computed in exponential time by Theorem 3.5. If \(s, t\) have sort \(L\), then, by Lemma 4.2, \(\text{List}(s), \text{List}(t)\) can be computed in polynomial time and the size of their elements is \(O(n)\) or \(O(t)\). Thus, the condition of Lemma 4.4 can be checked in exponential time by Theorem 3.5. Thus \(\text{list-check}(s, t)\) can be done in exponential time. The correctness follows from Lemmas 4.2 and 4.4 and Proposition 2.1.
Let $s, t \in T(\Sigma_{\text{max-min}+,s,t}, \mathcal{V})$ and $\delta_i \in \{\max, \min\}, v_i, v_i' \in T(\Sigma_{\text{max},s,t}, \mathcal{V})$ for $i = 1, \ldots, k$. Select a subterm occurrence of $\delta(u, v)$ in $s \approx t$ with $\delta \in \{\max, \min\}$ and $u, u' \in T(\Sigma_{\text{max},s,t}, \mathcal{V})$, and replace this occurrence with a fresh variable $z$—let the result be $s' \approx t'$. Let the translation $\mathbf{Pre}$ be replacing $\psi$ by $x_1 \approx \delta_1(u_1, v_1) \wedge \cdots \wedge x_k \approx \delta_k(u_k, v_k) \wedge z \approx \delta(u, v') \rightarrow s' \approx t'$. Clearly, the translation $\mathbf{Pre}$ is terminating and the final result $\mathbf{Pre}^\varphi(s)$ of the form $x_1 \approx \delta_1(u_1, v_1) \cdots \wedge x_k \approx \delta_k(u_k, v_k) \rightarrow s \approx t$ with $s, t \in T(\Sigma_{\text{max-min},s,t}, \mathcal{V})$. Next, replace each $x_i \approx \max(u_i, v_i)$ in $\mathbf{Pre}^\varphi(s)$ by $(u_i \leq v_i \rightarrow x_i \approx u_i)$ and $x_i \approx \min(u_i, v_i)$ in $\mathbf{Pre}^\varphi(s)$ by $(u_i \leq v_i \rightarrow x_i \approx u_i) \wedge (v_i < u_i \rightarrow x_i \approx v_i)$ for each $i = 1, \ldots, k$. Clearly, the result thus obtained is a PA formula and the translation preserves validity. Furthermore, the whole translation can be done in polynomial time. Thus, by the fact that the validity of universal PA formulæ[2] is decidable in exponential time, the following corollary is obtained.

**Corollary 5.2.** It is decidable in exponential time whether for given $s, t \in T(\Sigma_{\text{max-min},s,t}, \mathcal{V})$ the equation $s \approx t$ is an inductive theorem of $\mathcal{R}_{\text{max-min}}$.

In the rest of this subsection, we show another decision procedure for a subclass of the problems in the hope that the procedure performs more efficiently experimentally, albeit in exponential time complexity—namely, a decision procedure for the fragment without addition $+$ and successor function $s$. Thus, we present a decision procedure of $\mathcal{R}_{\text{max-min}} \models_{\text{id}} s \approx t$ for $s, t \in T(\Sigma_{\text{max-min}}, \mathcal{V})$, where $\Sigma_{\text{max-min}} = (\{N\}, \mathcal{F}_{\text{max-min}})$ with $\mathcal{F}_{\text{max-min}} = \{\max(x, y), \min(x, y), 0(0), 0\}$ and $\mathcal{R}_{\text{max-min}} = \{r \rightarrow r \in \mathcal{R}_{\text{max-min}} + t \in \mathcal{R}_{\text{max-min}}\}$.

Let $N_{\text{max-min}}(s, t) = \{N, \max^2, \min^2, s, 0\}$ be a $\Sigma_{\text{max-min}}$-algebra, defined in a similar way to $N_{\text{max-min},s,t}$. Note here that the successor function $s$ is necessary to construct $N_{\text{max-min},s,t}$.

**Lemma 3.5.** The initial $\Sigma_{\text{max-min}}$-algebra of $\mathcal{R}_{\text{max-min}}$ is isomorphic to $N_{\text{max-min}}$.

We are now going to present a decision procedure for $\mathcal{R}_{\text{max-min}} \models_{\text{id}} s \approx t$ for $s, t \in T(\Sigma_{\text{max-min}}, \mathcal{V})$, where $\Sigma_{\text{max-min}} = (\{N\}, \mathcal{F}_{\text{max-min}})$.

Let us denote by $P(X)$ the powerset of a set $X$.

**Procedure max-min-check** $(s, t)$

1. Normalize $s$ and $t$ by the following TRS: $S_{\text{max-min}} = \{\min(x, y, z) \rightarrow \max(x, z), \min(y, z)\}$

   $\min(x, y, z) \rightarrow \max(x, z), \min(y, z)\} = \{\min(x, y, z) \rightarrow \max(x, z), \min(y, z)\} \}$

   Note that $S_{\text{max-min}}$ is convergent and each term has a unique $S_{\text{max-min}}$-normal form. In this rest of the subsection, $s, t \in S_{\text{max-min}}$ is abbreviated as $s, t$.

2. We define (min)-terms and (max)-terms by the following BNF.

   (min)-terms $v_i ::= 0 \mid v \mid \max(v_1, v_2)$

   (max)-terms $w_i ::= 0 \mid v \mid \max(w_1, w_2)$

   where $x$ ranges over variables. Obviously, $s, t \in S_{\text{max-min}}$ are (max)-terms. For each (min)-term $v$, define a set $[v]_\text{min} \in \mathcal{P}(\mathcal{V}) \cup \{\top\}$ as follows: $[v]_\text{min} = \top$ if $v$ is a (min)-term containing $0$, and $[v]_\text{min} = \mathcal{V}(v)$, otherwise. For each (max)-term $w$, define a set $[w]_\text{max} \subseteq \mathcal{P}(\mathcal{V}) \cup \{\top\}$ as follows: $[w]_\text{max} = \{[v]_\text{min}\}$ for

   **Note that one has to check the validity of formulas of the form**

   $\forall x_1, \ldots, x_k. \forall \psi$ **for quantifier-free $\psi$ and not that of the form**

   $\exists x_1, \ldots, x_k. \psi$ **(satisfiability problem of quantifier-free PA formula),**

   which is in NP.
Clearly, $\rho$ holds. Thus it remains to show that max-min-check($s, t$) returns 'yes' iff $\exists \rho \Rightarrow s \approx t \Delta$. Suppose max-min-check($s, t$) returns 'yes'. Then, by definition of the procedure, $\rho$ holds. Thus it remains to show that max-min-check($s, t$) returns 'yes'. Hence, $\exists \rho \Rightarrow s \approx t \Delta$.

Thus, we first explain a corollary that follows from an existing work. In [1], an exponential decision procedure for $N$ is presented, where $N$ is isomorphic to $\Sigma_{\max(min)}$. We now briefly explain the decision procedure of [1]. Given an equation $s \approx t \Delta$ of $N$, normalize $s, t$ by the following TRS:

$$\begin{align*}
+(\max(x, y), z) & \rightarrow \max(+(x, z), +(y, z)) \\
+(z, \max(x, y)) & \rightarrow \max(+z, +(x, y)) \\
\max(x, 0) & \rightarrow x \\
\max(0, y) & \rightarrow y
\end{align*}$$

Let $s' \approx t'$ be an arbitrary but fixed result. Let (max-)contexts be given by the following BNF:

$$(\max)\text{-contexts} \quad \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
v_1 := \emptyset \mid \max(v_1, v_2)
\end{array}
\end{array}
\end{array}$$

Thus $s'$ and $t'$ are 0 or 0 and 0 is correct, so we suppose otherwise. Clearly, $C[u_1, \ldots, u_n] \subseteq C[t_1, \ldots, t_n]$ is valid if and only if $u_i \leq C[u_1, \ldots, u_n]$ is valid for all $1 \leq i \leq m$ and $u_i \leq C[u_1, \ldots, u_n]$ is valid for all $1 \leq j \leq n$. Thus, it suffices to decide the validity of inequalities $u \leq C[v_1, \ldots, v_m]$, for any (max-)context $C$ and $v_1, \ldots, v_m$ in $T(\Sigma_{\max})$. Let the formal polynomials of $u_i, v_1, \ldots, v_m$ be $v_1 = \sum_{i=1}^{m} c_i v_i$ and $\{ v_i \} = d_1 x_1 + \cdots + d_n x_n$ (1 $\leq m$). Corollary 3.17 of [1] shows that the inequation $u \leq C[v_1, \ldots, v_m]$ is valid if and only if there exist the non-negative real numbers $\lambda_1, \ldots, \lambda_m$ and $\gamma_1, \ldots, \gamma_n$ such that $\sum \lambda_i = 1$ and, for each $j = 1, \ldots, n$, $\sum \lambda_i d_j x_j = \gamma_j = 0$. The latter is a linear programming problem, which is known to be solvable in polynomial time.

Thus, the next corollary immediately follows from [1].

**Corollary 5.16.** It is decidable in exponential time whether for given $s, t \in T(\Sigma_{\max}), \nu$, the equation $s \approx t \Delta$ is an inductive theorem of $\mathcal{R}(\nu)$.

In the rest of this section, we show that by replacing addition $+$ with successor function $s$, one can obtain a polynomial decision procedure, i.e. $\mathcal{R}(\nu) \models_{\text{ind}} s \approx t$ by $s$ in polynomial time for $s, t \in T(\Sigma_{\max}), \nu$, where $\Sigma_{\max} = \{ \{ N \}, F(\max, s_0) \}$, $F(\max, s_0) = \{ \{ x \} \}, g(s_0, 0) \}$ and $\mathcal{R}(\nu) \models_{\text{ind}} s \approx t$ holds. Note that since $N_{\max+0}$ cannot encode $s$ and $N_{\max+0}$ cannot encode $s$. This result does not follow from Corollary 5.16.

**Lemma 5.17.** The initial $\Sigma_{\max, s_0}$-algebra of $\mathcal{R}(\nu)$ is isomorphic to $N_{\max, s_0}$.

We now present a decision procedure for the validity on $N_{\max, s_0}$.

**Procedure $\max$-check($s, t$)**

1. Normalize $s, t$ by the following TRS:

$$\begin{align*}
S(\max) = & \{ s(x, y) \rightarrow \max(s(x), s(y)) \}
\end{align*}$$

Note that $S(\max)$ is isomorphic to $\nu$ and each term has a unique $S(\max)$-normal form. In the rest of this subsection, $s \Delta \nu = s \Delta$. We define (s)-terms and (max)-terms by the following BNF:

$$(\max)\text{-terms} \quad \begin{array}{c}
\begin{array}{c}
v_1 := v_1 \mid \max(v_1, v_2)
\end{array}
\end{array}$$

where $x$ ranges over variables. Obviously, $s, t \Delta \nu$ are (max)-terms. From each (max)-term $\nu$, we define a set $\nu$ of (s)-terms by $s \Delta \nu = \{ s \}$ if $u$ is an (s)-term, $\{ s(v_1, v_2) \} = \{ v_1 \} \cup \{ v_2 \}$ otherwise. Compute the sets $U = \{ s \Delta \nu \}$ and $V = \{ t \Delta \nu \}$ of (s)-terms.

3. Define a relation $\prec$ on (s)-terms by $s \prec t$ if either (1) $s(x) \prec s(x)$, or (2) $s(x) \prec t(x)$, for some $u \in V \cup \{ 0 \}$, for (1) $s(x) \prec t(x)$, for some $u \in V \cup \{ 0 \}$, and $x \in V$. For each set $X$ of (s)-terms, let $\max(X)$ be the set of maximal elements of $X$ with $\prec$.

$$\max(X) = \{ s \in X \mid s \prec t \text{ for no } t \in X \}$$

Compute the sets $S = \max(U)$ and $T = \max(V)$ of (s)-terms. Then return 'yes' if $S \equiv T$ and 'no' otherwise.

**Lemma 5.18.** $\max$-check($s, t$) can be done in polynomial (quadratic) time.

**Lemma 5.19.** For any $s, t \in T(\Sigma_{\max, s_0}, \nu)$, $\max$-check($s, t$) returns 'yes' iff $N_{\max, s_0} \models s \approx t \Delta$.

**Theorem 5.20.** It is decidable in polynomial time whether for given $s, t \in T(\Sigma_{\max, s_0}, \nu)$ the equation $s \approx t \Delta$ is an inductive theorem of $\mathcal{R}(\nu)$.

**Example 5.21.** Let $s = \max(s(x), x)$ and $t = \max(s(0), s(x))$. Then $\max$-check($s, t$) returns 'yes', as we have $\max(\nu) = \max(\{ s(x), s(x) \}) = \{ s(x) \}$ and $\max(\nu) = \max(\{ s(0), s(x) \}) = \{ s(x) \}$. Thus, we conclude from Theorem 5.20 that $\mathcal{R}(\nu) \models_{\text{ind}} s \approx t \Delta$.
It is not difficult to give a decision procedure and obtain the next theorem in the similar way to Theorem 5.22.

**Theorem 5.22.** It is decidable in polynomial time whether for given \( s, t \in T(\Sigma_{(\text{min},+),0}, V) \) the equation \( s \approx t \) is an inductive theorem of \( R_{(\text{min})} \).

### 6. Experiments

The decision procedures of the paper have been implemented in the SPIKE prover ⁴[28].

For the experiments, we take into account some categories and construct collections of conjectures randomly for each category. Each collection gathers equalities of same inductive validity and built over one of the following signatures:

- \( \Sigma_{(+,0)} \) and its extension with \( 0, \Sigma_{(+,+,0)} \).
- \( \Sigma_{(x,+,s)} \) and its extension with \( 0, \Sigma_{(x,+,+,0)} \).
- \( \Sigma_{(x,+,s)} \) and its extension with \( 0, \Sigma_{(x,+,+,s)} \).
- \( \Sigma_{(\text{len}rev,\text{rev},\text{rev},+,+,\text{nil},\text{nil},s,0)} \) by considering equalities between terms of list sort,
- \( \Sigma_{(\text{max},\text{min},0)} \),
- \( \Sigma_{(\text{max},s,0)} \) and similarly \( \Sigma_{(\text{min},s,0)} \).

For each category, we excluded trivial equations and equations whose roots of both sides of the equation is \( s \). All equalities have at most three distinct variable of each sort and the depth of both sides is smaller than five. The number of examples widely varies between one category.

Furthermore, since most of the randomly generated conjectures are not inductive theorems, we have additionally incorporated several ad-hoc heuristics to generate a sufficient number of inductive theorems to reach a target of 100 examples (we failed for only one category).

The numbers of examples and the summary of experiments are shown in Table 1. Each test was performed on a PC with one 2.50GHZ CPU and 4G of memory. For any information of the form \( a(b) \) in the table, \( a \) (resp. \( b \)) represents the number of examples that has (resp. has not) been successfully checked within the 10 seconds.

Our decision procedures successfully solved all these examples, as shown in the column entitled “SPIKE + direct”. For comparison, we also tested the examples with SPIKE + integrating an incomplete solver for PA, as previously described in ⁴[29]. The figures from the columns entitled “SPIKE + PA (Cor. 5.2)” (resp., “SPIKE + PA (4)”) give the statistics about the use of the PA solver with a prior encoding of equalities to PA, according to Corollary 5.2 (resp., without encoding). In the categories to which our decision procedures and the PA solver are applicable, the first are 2–6 times faster; the latter failed at some examples due to lack of additional resources.

When the decision procedures and the PA solver are disabled, SPIKE acts as an implicit induction theorem prover, able to perform several induction and rewrite steps during a proof session. We used a unique proof strategy for all tests and no additional lemmas (0-knowledge proofs). Apart from the TRS and the conjectures to be proved, we additionally provided a unique precedence suitable for ensuring the termination of the input TRS. For some categories, SPIKE inductively proved disproved most of the examples, as shown in the column “SPIKE (induction) [28]”. However, for some categories more than half of the examples have not been solved or require more than 10s to be solved. A special category, involving equations over the extended signature \( \Sigma_{(\text{max},\text{min},+,+,s,0)} \) helped to better compare with the PA solver. ‘-’ means that the TRS category cannot be solved by the corresponding SPIKE configuration, hence it matches the values from the last column. We can safely conclude that SPIKE has become more effective by incorporating our decision procedures.

The collection of examples and details of the experiments are available on the webpage [http://www.nue.riec.tohoku.ac.jp/tools/experiments/ppdp14/](http://www.nue.riec.tohoku.ac.jp/tools/experiments/ppdp14/)

### 7. Conclusion

We have given decision procedures for checking the inductive validity of equations built over common function symbols defined on natural numbers and lists. Our results are summarized in Table 2 In contrast to the line of research from ¹[11,14], our decidability results do not impose any syntactical conditions on the equations or induction reasoning albeit specific to some TRSs. Experiments show that our decision procedures are effective for enhancing inductive theorem provers.

Our strong restriction on TRSs can be slightly relaxed. For example, the decidability result for \( R_{(\text{min},+)} \) also applies to the following variation \( R_{(\text{min},+)}' \), which is convergent, sufficiently complete and has the same initial algebra as \( R_{(\text{min},+)} \):

\[
R_{(\text{min},+)}' = \begin{cases} 
+ (x, 0) & \rightarrow x \\
+ (x, s(y)) & \rightarrow + (s(x), y) \\
\times (x, 0) & \rightarrow 0 \\
\times (x, s(y)) & \rightarrow + (x, \times (x, y)) 
\end{cases}
\]

As future works, we would like to tackle some problems which are left open in the current contribution: extending the decision procedure of ¹[11] to equations over \( \Sigma_{(\text{max},+,0)} \) and to equations over \( \Sigma_{(\text{min},+,0)} \), obtaining a decision procedure for equations over \( \Sigma_{(\text{max},+,s,0)} \). We also intend to find other standard TRSs for which our approach works, and how to (semi-)automatically find inductively valid equations that sufficiently characterize the validity in initial algebras. We also intend to apply our approach for classes of conditional TRSs and (conditional) equations.

### Acknowledgements

Thanks are due to Yoshihito Toyama for kindly allowing us to include his unpublished result ⁸[30]. We also thank Yuki Chiba, Nao Hirokawa and Michio Oyamaguchi for helpful comments. We are also grateful to our anonymous reviewers for their time and comments.

### References

8. R. S. Boyer and J. S. Moore. *Integrating decision procedures into heuristic theorem provers: a case study of linear arithmetic*. In *Ma-
Table 1. Summary of Experiments

<table>
<thead>
<tr>
<th>TRS</th>
<th>signature / complexity</th>
<th>inductive validity</th>
<th>number of problems</th>
<th>SPIKE + direct</th>
<th>SPIKE + PA(Cor. 5.2)</th>
<th>SPIKE + PA(1)</th>
<th>SPIKE (induction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{R}(+)$</td>
<td>$\Sigma_{(+,s)}$ / polynomial</td>
<td>yes</td>
<td>101</td>
<td>101(0)</td>
<td>101(0)</td>
<td>101(0)</td>
<td>30(71)</td>
</tr>
<tr>
<td>$\mathcal{R}(+)$</td>
<td>$\Sigma_{(+,s,0)}$ / polynomial</td>
<td>no</td>
<td>1197</td>
<td>1197(0)</td>
<td>1197(0)</td>
<td>1197(0)</td>
<td>887(310)</td>
</tr>
<tr>
<td>$\mathcal{R}(x,+)$</td>
<td>$\Sigma_{(x,+),s}$ / exponential</td>
<td>yes</td>
<td>101</td>
<td>101(0)</td>
<td>101(0)</td>
<td>101(0)</td>
<td>75(26)</td>
</tr>
<tr>
<td>$\mathcal{R}(x,+)$</td>
<td>$\Sigma_{(x,+),s,0}$ / polynomial</td>
<td>no</td>
<td>536</td>
<td>536(0)</td>
<td>536(0)</td>
<td>536(0)</td>
<td>407(129)</td>
</tr>
<tr>
<td>$\mathcal{R}(x,+)$</td>
<td>$\Sigma_{(x,+),s,0}$ / exponential</td>
<td>yes</td>
<td>60</td>
<td>60(0)</td>
<td>-</td>
<td>-</td>
<td>21(38)</td>
</tr>
<tr>
<td>$\mathcal{R}(x,+)$</td>
<td>$\Sigma_{(x,+),s,0}$ / exponential</td>
<td>no</td>
<td>1670</td>
<td>1670(0)</td>
<td>-</td>
<td>-</td>
<td>992(678)</td>
</tr>
<tr>
<td>$\mathcal{R}($len,rev,$\delta,x,+)$</td>
<td>$\Sigma_{(len,rev,\delta,x,+),s,0}$ / exponential</td>
<td>yes</td>
<td>244</td>
<td>244(0)</td>
<td>-</td>
<td>-</td>
<td>145(199)</td>
</tr>
<tr>
<td>$\mathcal{R}($len,rev,$\delta,x,+)$</td>
<td>$\Sigma_{(len,rev,\delta,x,+),s,0}$ / exponential</td>
<td>no</td>
<td>1204</td>
<td>1204(0)</td>
<td>-</td>
<td>-</td>
<td>910(294)</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min,+)$</td>
<td>$\Sigma_{(\max,\min,+,s,0)}$ / polynomial</td>
<td>yes</td>
<td>213</td>
<td>213(0)</td>
<td>-</td>
<td>-</td>
<td>241(89)</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min,+)$</td>
<td>$\Sigma_{(\max,\min,+,s,0)}$ / exponential</td>
<td>no</td>
<td>1777</td>
<td>1777(0)</td>
<td>-</td>
<td>-</td>
<td>1068(709)</td>
</tr>
<tr>
<td>$\mathcal{R}(\max)$</td>
<td>$\Sigma_{(\max,s,0)}$ / polynomial</td>
<td>yes</td>
<td>153</td>
<td>153(0)</td>
<td>148(5)</td>
<td>-</td>
<td>121(32)</td>
</tr>
<tr>
<td>$\mathcal{R}(\max)$</td>
<td>$\Sigma_{(\max,s,0)}$ / polynomial</td>
<td>no</td>
<td>1528</td>
<td>1528(0)</td>
<td>1478(50)</td>
<td>-</td>
<td>1285(243)</td>
</tr>
<tr>
<td>$\mathcal{R}(\min)$</td>
<td>$\Sigma_{(\min,s,0)}$ / polynomial</td>
<td>yes</td>
<td>116</td>
<td>116(0)</td>
<td>116(0)</td>
<td>-</td>
<td>114(2)</td>
</tr>
<tr>
<td>$\mathcal{R}(\min)$</td>
<td>$\Sigma_{(\min,s,0)}$ / polynomial</td>
<td>no</td>
<td>1107</td>
<td>1107(0)</td>
<td>1107(0)</td>
<td>-</td>
<td>1061(46)</td>
</tr>
</tbody>
</table>

Table 2. Summary of Results

<table>
<thead>
<tr>
<th>TRS</th>
<th>signature</th>
<th>complexity</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{R}(\exp,x,+)$</td>
<td>$\Sigma_{(\exp,x,+),s,0}$</td>
<td>decidable</td>
<td>(10)</td>
</tr>
<tr>
<td>$\mathcal{R}(\eq,x,+)$</td>
<td>$\Sigma_{(\eq,x,+),s,0}$</td>
<td>undecidable</td>
<td>(30)</td>
</tr>
<tr>
<td>$\mathcal{R}(x,+)$</td>
<td>$\Sigma_{(x,+),s,0}$</td>
<td>exponential</td>
<td>Thm. 5.5</td>
</tr>
<tr>
<td>$\mathcal{R}(\max)$</td>
<td>$\Sigma_{(\max,s,0)}$</td>
<td>polynomial</td>
<td>Thm. 5.7</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min)$</td>
<td>$\Sigma_{(\max,\min,+,s,0)}$</td>
<td>exponential</td>
<td>Thm. 5.8</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min,\delta,\max)$</td>
<td>$\Sigma_{(\max,\min,\delta,\max,+,s,0)}$</td>
<td>exponential</td>
<td>Thm. 5.14</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min,\delta,\max)$</td>
<td>$\Sigma_{(\max,\min,\delta,\max,+,s,0)}$</td>
<td>exponential</td>
<td>Thm. 5.17 of (11)</td>
</tr>
<tr>
<td>$\mathcal{R}(\max,\min)$</td>
<td>$\Sigma_{(\max,\min,+,s,0)}$</td>
<td>polynomial</td>
<td>Thm. 5.20</td>
</tr>
</tbody>
</table>


A. Proofs

Proof: [of Lemma 3.1] First, it is easy to check that \(R_{(x,+)}\) is convergent and sufficiently complete \(\mathcal{S}\). Hence any ground term \(t \in T(\Sigma_{(x,+),0})\) has a unique normal form \(t_\downarrow \in T(\Sigma_{(x,+),0})\). Thus the mapping given by \(\varphi: [t] \mapsto t_\downarrow\) is a bijection from \(T(\Sigma_{(x,+),0})/\sim_{R_{(x,+)}}\) to \(T(\Sigma_{(x,+),0})\).

Proof: [of Lemma 3.2] By induction on \(s\).

Proof: [of Lemma 3.3] Suppose \(N_{(x,+),0} = s \approx t\). Then, for any valuation \(\rho: V \rightarrow N\), we have \([s]^{\rho}_{N_{(x,+),0}} = [t]^{\rho}_{N_{(x,+),0}}\). Since \(N_{(x,+),0}\) is a subalgebra of \(Z^\ast\), it follows that, for any valuation \(\rho: V \rightarrow N\), \([s]^{\rho}_{N_{(x,+),0}} = [t]^{\rho}_{N_{(x,+),0}}\). Hence by Lemma 3.2, \([s]^{\rho}_{N_{(x,+),0}} = \approx [t]^{\rho}_{N_{(x,+),0}}\) for any \(a_1, \ldots, a_n \in N\).

Proof: [of Theorem 3.7] As for Theorem 3.3 but taking into account that the computation of the list of monomial expressions \(\text{Mono}(s)\) can be done in \(O(|s|)\) if the term \(s\) does not contain the multiplication \(\times\).

Proof: [of Proposition 3.8] Let us denote by \(Z_\ast[x_1, \ldots, x_n]\) the set of formal polynomials over the indeterminates \(x_1, \ldots, x_n\) with natural number coefficients. First note that \(3a_1, \ldots, a_n \in Z_\ast[\varphi(a_1, \ldots, a_n) = 0\) if \(3a_1, \ldots, a_n \in N\, \varphi(\delta a_1, \ldots, \delta a_n) = 0\) for some \(\delta a_1, \ldots, \delta a_n \in \{1, -1\}\).

Proof: [of Lemma 4.1] Similar to Lemma 3.1. Note that it is easy to check that \(R_{(\text{len},\text{rev},0,+)}\) is convergent and sufficiently complete \(\mathcal{S}\).

Proof: [of Lemma 4.2] Let \(s \in T(\Sigma_{(\text{single},+),0})\). First, eliminate all occurrences of \(\uparrow\) by the third rule. The size of the term is at most doubled by this. Then, normalize recursively using len-rules the outermost subterms of the form \(\text{len}(u)\) with \(u \notin V\). Next, normalize using rev-rules by counting the number \(k\) of occurrences of rev along the path, swapping arguments of \(\uparrow\) if \(k\) is odd and eliminating rev. Finally, if \(s\) has sort \(N\) replace each \(\text{len}(x)\) with \(\delta(\text{len}(x))\), i.e. (i.e. if \(s\) has sort \(L\) change the final \((\text{list})\)-term to a list of \((\text{elem})\)-terms by listing up all non-nil \((\text{elem})\)-terms from left to right. Clearly, \(\text{Abs}(s_L)\) (or \(\text{List}(s_L)\)) is obtained, and each step can be done in \(O(|s|)\). Thus overall computation can be done in polynomial time.

Proof: [of Lemma 4.3] We show (1) and (2) simultaneously by Noetherian induction on \(s\) w.r.t. \(\Delta \cup \rightarrow_S\).

1. \(s \in V^L\). Then \(s_L = (s)\) is an \((\text{elem})\)-term, which is a (degenerated) \((\text{list})\)-term.

2. \(s = \text{nil}\). Then \(s_L = (s)\) is a \((\text{list})\)-term.

3. \(s := ((t_1, t_2)).\) Then \(s_L := (\text{single}(t_1, t_2)) = s'\). By induction hypothesis, \(t_1\downarrow\) and \(t_2\downarrow\) are \((\text{list})\)-terms, so \(s_L := (s')\) is a \((\text{list})\)-term.

4. \(s = \text{single}(t).\) Then, by induction hypothesis, \(t_\downarrow \in T(\Sigma_{(x,+),0}, V^N \cup \{\text{len}(x)\} \cup \{x \in V^L\})\). Since \(s_L = \text{single}(t), s_L\) is a \((\text{list})\)-term by definition.

5. \(s = \text{rev}(t).\) Then by induction hypothesis \(t_s\downarrow \in \text{a \((\text{list})\)-term.}\)

6. \(s = \text{rev}(t). \) Then by induction hypothesis \(t_s\downarrow \in \text{a \((\text{list})\)-term.}\)

We distinguish two cases.

(a) \(w_1 = \text{nil} \land w_2 = \text{nil}\). Then \(s := s' \rightarrow_S (w_1, w_2) \rightarrow_S s'\) for some \(s'\). The claim follows from \(s_L = s'\downarrow\) and the induction hypothesis.

(b) Otherwise, for \(i = 1, 2\), either \(w_i = (\delta(w'_i, w'_i))\), \(w_i = x \in V^L\), \(w_i = \text{single}(u_i)\) for some \(u_i \in T(\Sigma_{(x,+),0}, V^N)\), or \(w_i = \text{rev}(x)\). In either cases, \(\delta(w_1, w_2)\) is a \((\text{list})\)-term.
Proof: [of Lemma 4.4] For any \( l \rightarrow r \in S \) and valuation \( \rho \) on \( L \), \([ l \rho ] = [ r \rho ]\). Hence \( L \models s \equiv s \) and \( L \models t \equiv t \). Thus \( L \models s \oplus t \), i.e. \( L \models s \equiv t \). Let \( s, t \in T(\Sigma_{\{a, \lnot \}}) \). Then \( L \models \Sigma_{\{a, \lnot \}} \).

Since the length of lists ranges over natural numbers, for any valuation \( \rho \) on \( N \), there exists a valuation \( \rho' \) on \( L \) such that \([ A \downarrow ]_{\rho'} \) and \([ A \downarrow ]_{\rho} \). Thus, it follows from Lemma 4.3 and \( T(S) \subseteq T(S_{\{a, \lnot \}}) \), i.e. \( T(\Sigma_{\{a, \lnot \}}) \). That is, \( \Sigma_{\{a, \lnot \}} \).

Thus, the overall computation can be done in \( \mathcal{O}(2^{(|s|+|t|)}) \).

Proof: [of Lemma 5.10] By induction on \( w \), we have the following cases:

1. \( w \) is a (min)-term. Then, using Lemma 5.8
2. \( \forall \rho \), \( \exists w' \)
3. \( \forall \rho \), \( \exists w' \)
4. \( \forall \rho \), \( \exists w' \)
5. \( \exists \rho \), \( \forall \rho \)
6. \( \exists \rho \), \( \forall \rho \)

Proof: [of Lemma 5.11] Suppose \( A, B \in S \) and \( \exists A \neq B \neq A \). Then by Lemma 5.7, \( \ominus \rho \). Thus \( \forall \rho \).

Proof: [of Lemma 5.12] Obvious. (\( \equiv \)) Let \( X = \{ x \in \mathbb{N} \mid 3x \neq \exists x \in S \cup \mathbb{N} \} \). Suppose \( \forall \rho \). Then we have \( \exists \rho \), \( \forall \rho \).

Proof: [of Theorem 5.14] For the correctness of \( \mathcal{O}(\Sigma_{\{a, \lnot \}}) \).

Proof: [of Lemma 5.5] We distinguish six cases, where we omit the symmetric case.

1. \( u_1 = x \) and \( v_1 = y \) with \( x \neq y \). Then it’s easy to see \([ u_1 \rho ] \neq [ v_1 \rho ] \) for some \( \rho \).
2. \( u_1 = x \) and \( v_1 = \text{single}(u) \). Then it suffices to take \( \rho \) such that \( \rho(x) = [1] \).
3. \( u_1 = x \) and \( v_1 = \text{rev}(y) \). If \( x \neq y \) then it’s easy to see \([ u_1 \rho ] \neq [ v_1 \rho ] \) for some \( \rho \).
4. \( u_1 = \text{single}(u) \) and \( v_1 = \text{single}(v) \). Then we have \( u_1 \neq v_1 \) and hence \([ u_1 \rho ] \neq [ v_1 \rho ] \) for some \( \rho \) by the case (1). This implies \([ u_1 \rho ] \neq [ v_1 \rho ] \) for some \( \rho \).
5. \( u_1 = \text{single}(u) \) and \( v_1 = \text{rev}(x) \). Then it suffices to take \( \rho \) such that \( \rho(x) = [1] \).

Proof: [of Theorem 5.7] Similarly to the proof of Theorem 4.5 using Theorem 3.7 instead of Theorem 3.5.
$s \approx t$ iff $N_{\text{max}, \text{min}, s, t} = s \approx t$ iff $\text{max-min-check}(s, t)$ returns ‘yes’. By Lemma 5.17 the procedure $\text{max-min-check}(s, t)$ runs in exponential time. □

**Proof:** of Lemma 5.17 Similar to Lemma 5.1. □

**Proof:** of Lemma 5.18 Procedure $\text{max-check}$ can be implemented more efficiently like this:

1. For each occurrence of $u \in V \cup \{0\}$ in $s$, count the number $k$ of $s$ from the root of the term to that occurrence, and collect all $(k, u)$. This can be done in $O(|s|)$. Let the collection be $S$. Clearly, $|S| \leq |s|$. Compute a set $T$ by applying the same procedure to $t$.

2. Eliminate non-maximal elements w.r.t. $\prec$ in $S(T)$. This can be done in $O(|s|)$ (resp. $O(|T|)$).

3. Check $S = T$ (as sets of $(s)$-terms). This can be done in $O(|S| \times |T|)$.

Thus the overall procedure is $O((|s| + |t|)^2)$. Hence the claim follows. □

**Proof:** of Lemma 5.19 In the following, let us abbreviate $N_{\text{max}, \text{min}, s, t} = N$ as $N$, $u \downarrow$ as $u_\downarrow$ for $u \in \{s, t\}$. Since $N \models s(x, y) \approx \text{max}(s(x), s(y))$, we have $N \models s \approx t$ iff $N \models s \approx t$. Thus it remains to show that $\text{max-check}(s, t)$ returns ‘yes’ iff $N \models s \approx t$.

For any non-empty finite set $U = \{u_1, \ldots, u_n\}$ of $(s)$-terms, let $\text{Max}(U) = \text{max}(u_1, \text{max}(u_2, \ldots, \text{max}(u_{n-1}, u_n)) \ldots))$. Then $N \models \text{Max}(u) \approx \text{Max}(u)$ for $u \in \{s, t\}$ by definition.

For any $u = s^n(z)$ and $v = s^n(z)$ with $n \leq m (z \in V \cup \{0\})$, we have $N \models \text{Max}(u, v) \approx v$. Thus for any $(s)$-terms $u, v$ such that $u \prec v$, we obtain $N \models \text{Max}(u, v) \approx v$.

Hence, we have $N \models \text{Max}(\text{Max}(U)) \approx \text{Max}(U)$ for any non-empty finite set $U$ of $(s)$-terms. Hence we have $N \models u_\downarrow \Rightarrow \text{Max}(\{u_\downarrow\}) \approx \text{Max}(\{u_\downarrow\})$ for $u \in \{s, t\}$ by definition. Let $S = \text{Max}(\{s_\downarrow\})$ and $T = \text{Max}(\{t_\downarrow\})$.

Suppose $\text{max-check}(s, t)$ returns ‘yes’. Then, by the definition of the procedure, we have $S = T$. Then, $N \models s_\downarrow \approx \text{Max}(S) \approx \text{Max}(T) \approx t_\downarrow$, thus $N \models s_\downarrow \approx t_\downarrow$. Thus it remains to show the converse.

Suppose $N \models s_\downarrow \approx t_\downarrow$. By $N \models s_\downarrow \approx \text{Max}(S)$ and $N \models t_\downarrow \approx \text{Max}(T)$, we have $N \models \text{Max}(S) = \text{Max}(T)$. By the definition of the procedure, it suffices to show $S = T$.

Let $X = \{x \in V \mid s^n(x) \in S \cup T \text{ for some } n \in N\}$ and $Y = \{x \in V \mid s^n(x) \in T \text{ for some } n \in N\}$. Let $\rho = \{x \mapsto 0 \mid x \in X \cup Y\}$ and $k = \text{max}([u]_\rho \mid u \in S \cup T)$.

Firstly, we claim $X = Y$. For this, we suppose $X \neq Y$ and show the contradiction. W.l.o.g. assume $s^n(x) \in S$ and $x \notin Y$. Then by taking a valuation $\delta = \{x \mapsto k + 1\} \cup \{y \mapsto 0 \mid x \neq y, y \in X \cup Y\}$, we have $[\text{Max}(S)]_\delta \geq k + n + 1 > k = [\text{Max}(T)]_\delta$. This is a contradiction. Thus we have $X = Y$.

Next we show that if $s^n(x) \in S$ and $s^{m}(x) \in T$ then $n = m$. Suppose to the contrary that there exists $x \in V$ such that $s^n(x) \in S$ and $s^m(x) \in T$ with $n \neq m$. Then by taking $\delta = \{x \mapsto k + 1\} \cup \{y \mapsto 0 \mid x \neq y, y \in X \cup Y\}$, we have $[\text{Max}(S)]_\delta = n + k + 1 \neq m + k + 1 = [\text{Max}(T)]_\delta$. This is a contradiction. Thus $s^n(x) \in S$ and $s^m(x) \in T$ imply $n = m$.

Let $U = \{s^n(x) \mid s^n(x) \in S, x \in V\}$. We have shown that either $S = U$ or $S = U \cup \{s^0(0)\}$ and either $T = U$ or $T = U \cup \{s^m(0)\}$ for some $n, m$. If $S = U = T$, we are done. Suppose $S = U$ and $T = U \cup \{s^m(0)\}$. Then, by definition of $\prec$, we have $m > n$ for any $s^n(x) \in U$. Thus, by taking a valuation $\rho = \{x \mapsto 0 \mid x \in X\}$, $[\text{Max}(S)]_\rho = \text{max}(n \mid s^n(x) \in U) < m = [\text{Max}(T)]_\rho$. This is a contradiction. Similarly, it does not happen the case $S = U \cup \{s^0(0)\}$ and $T = U$ for some $n$. It remains to show that if $S = U \cup \{s^0(0)\}$ and $T = U \cup \{s^m(0)\}$ then $l = m$. Suppose $l \neq m$. Suppose $S = U \cup \{s^0(0)\}$ and $T = U \cup \{s^m(0)\}$. Then by definition of $\prec$, we have $l, m > n$ for any $s^n(x) \in U$. Thus, by taking a valuation $\rho = \{x \mapsto 0 \mid x \in X\}$, $[\text{Max}(S)]_\rho = \text{max}(n \mid s^n(x) \in U) < m = [\text{Max}(T)]_\rho$. This is a contradiction. □