A New Method for Undecidability Proofs of First Order Theories

by

Ralf Treinen

A 09/90

Saarbrücken, May 1990

Ralf Treinen
Fachbereich 14 - Informatik
Im Stadtwald
Universität des Saarlandes
D6600 Saarbrücken

Electronic Mail: treinen@cs.uni-sb.de

Contents

1 Introduction

The interest of this paper is twofold. First it proposes a general methodology for proving results of the kind:

> The first order theory of the predicate logic model $\mathcal{I} = \cdots$ is undecidable.

Second, besides examples that serve just for the illustration of the method proposed, we show some applications that are interesting in their own right.

We only consider theories of given models, in contrast to theories defined by some sets of axioms that are not necessarily complete. When applied to (the theory of) a given model $\mathcal{I}$, the method leads to an effective mechanism that yields for each Post Correspondence Problem P over the alphabet $\{a, b\}$ a formula, denoted $\underline{\texttt{solvable}}_P$, such that

$$P \text{ is solvable} \quad \Leftrightarrow \quad \mathcal{I} \models \underline{\texttt{solvable}}_P \tag{1}$$

Because of the effectiveness of the construction of this formula we immediately get the undecidability result for the theory from the well-known undecidability of Post's Correspondence Problem. Furthermore we are interested in showing not only undecidability of the whole theory of $\mathcal{I}$, but of a smallest possible fragment

of this theory. In the construction of $\underline{\texttt{solvable}}_P$ we will therefore try to avoid alternations of quantifiers as far as possible.

The basic principle of the proof method proposed is the simulation of the two data types involved in Post's Correspondence Problem: strings and sequences (resp. sets). The representation of the objects of these data types is performed by appropriate representation functions mapping the carrier sets into the universe of the model under consideration. The representation does not reflect directly in the theory of the model, especially there is no need for formulas characterizing the images of the representation functions. The operations on the data types are expressed by first order formulas that are to be designed in regard to the properties of the model.

The target formula $\underline{\texttt{solvable}}_P$ consists of a "frame" that is independent of the model under consideration but uses subformulas representing the operations on the data types. We present the frame formula and formulate the requirements that guarantee the "correctness" of the representation of the carrier sets and the pertaining operations.

To a large extend we constrain the meaning of the formulas only for those elements of the universe that represent objects of the data types. Moreover, beyond the correctness of data type representation we have to make sure that a certain relation on the universe is Noetherian. This is an inherent property of the model, since the well-foundedness of a relation is not expressible in first order logic[1].

Several other methods for proving undecidability of theories have been proposed in the literature. [Tar53] shows that a theory T is undecidable if some essentially undecidable and finite axiomatizable theory T' (for instance the theory Q [TMR53]) is relatively weakely interpretable in T. In order to show relative weak interpretability of T' in T one has to find first order formulas defining the universe and operations of T' in some consistent extension of T. Hence the correspondence between the theories is expressed completely within the logic.

The method of [Rab65] does not require a finite axiomatization of the underlying undecidable theory. In its basic form it reads: A theory T is undecidable if there are an undecidable theory T' and T-formulas representing the universe, predicates and functions of T such that it is possible to establish a correspondence between the models of T and T'. In this way the translation of T' into T is again expressed in terms of first order logic, but in order to show the correctness of the translation it is necessary to prove the required corresponence of models.

The method proposed here takes a different point of view: It exploits the properties of the model instead of properties of the *theory of* the model. The logic is not involved in the definition of the representation functions: The intended applications concern universes constituting a formal language, such that the rep-

[1]Not even by a sentence of the extension $L_{\omega_1,\omega}$ containg countably infinite disjunctions and conjunctions ([Kei71]).

resentation may often be performed on a purely symbolic level. The logic is only used in the realization of the operations of the data types. We will demonstrate some applications where this technique yields very simple reduction proofs.

A first set of applications illustrating the method proposed is concerned with equational problems, that is validity of formulas with equality as the only predicate symbol in the initial, respectively the free algebra of an equational specification (see [BS89] for an introduction into equational problems). The first example (A) treats the decision problem for the theory of ground term algebras modulo the axioms of associativity and commutativity (AC for short) and has been given as open problem in [Com88]. In this paper the existential fragment has been shown decidable thus extending the results for AC unification ([Sti81] and [Kir85] for the case of additional free function symbols). The generalization of this result to the free algebra refutes a conjecture of [BS89] where general equational problems are claimed to be decidable provided unification with free function symbols is decidable. The extension by the axiom of idempotency to AC1 in Example (B) is straightforward.

To the authors knowledge the undecidability of the theory of ground terms modulo associativity alone (Example (F)) has not been stated before, but since Quine ([Qui46]) has proved the undecidability of the theory of concatenation this is not a surprising result. Unification modulo associativity has been shown to be decidable by Plotkin ([Plo72]). In contrast to the AC case, associativity without commutativity is of unification type ω (see [BHS86] for the classification of unification problems), this coincides with the observation that our technique yields undecidability of the Σ_3 fragment in the AC case but Σ_2 in the case of associativity.

The second field of application is the theory of ground terms equipped with some ordering relation. The undecidability of the "theory of subterm relation" has been shown in [Ven87] but without the extension to possibly infinite trees. Furthermore [Ven87] shows the decidability of the existential fragment. We mention this application in order to illustrate the benefit gained from a systematic study of reduction proofs.

The question of decidability of the theory of a total simplification ordering has been posed in [Com88]. The decidability of the existential fragment of a total lexicographic path ordering (lpo for short) is shown in [Com90b]. We prove in Example (C) the undecidability of the Σ_4 fragment of a partial lpo. Unfortunately there still remain two big gaps between these results (see Section 5).

The undecidability of the Σ_2 fragment of complete number theory (Example (G)) is of course by no means a new result; it is presented here merely for demonstrating some aspects of the method proposed. The undecidability of the Σ_1 fragment has been shown in [Mat70].

The separation of Post's Correspondence Problem into two datatypes induces the structure of the paper: After a survey of the mathematical framework in Section 2

the simulation of the data type "strings" is discussed in Section 3. In the applications this part will always be the trivial one. Section 4 describes the construction of the sentence $\underline{\texttt{solvable}}_P$ while presenting two alternative methods for the representation of construction sequences. In the first method sequences are viewed as sets. This method is easier to use than the second one representing sequences directely but is less powerful. On the other hand in some applications the second method can yield a smaller number of quantifier alternations in the formula $\underline{\texttt{solvable}}_P$.

2 Preliminaries

In this paper we consider unsorted first order logic where equality is not required. For the basic notions according syntax and semantics of first order logic the reader is referred to textbooks on mathematical logic, for instance [End72]. We specify a predicate logic basis as a pair (P, F) where the set of function symbols F is given in the form $\langle f(n_f), g(n_g), \ldots \rangle$ and the set of predicate symbols $P = \langle \oplus(n_\oplus), \odot(n_\odot), \ldots \rangle$. The numbers in parantheses are not part of the syntax but indicate the arity of the symbols. If $=(2)$ is present in P it is always interpreted as equality. We will frequently use symbolic names for formulas, and in defining one formula we will often refer to other formulas via their symbolic "macro" names without giving an exact semantics of macro expansion for formulas. We only mention the following notions adopted from [CK73]:

$w(x_1, \ldots, x_n)$ where w is a symbolic name for a formula stands for a formula the free variables of which are (possibly as proper subset) among $\{x_1, \ldots, x_n\}$. The main purpose of this notion is to fix an order on the "formal parameters" of a formula which simplifies notation with regard to instantiations. Once an order of parameters is established we can define $w(t_1, \ldots, t_n)$ as the formula obtained by simultanously replacing each "formal parameter" x_i by the corresponding "actual parameter" t_i. It is understood that bounded variables in w are renamed in such a way that no free variable of the t_i is captured by a quantifier of w. We write for elements $r_1, \ldots, r_n$ from the universe of the model $\mathcal{I}$: $\mathcal{I} \models w[r_1, \ldots, r_n]$ iff w is satisfied in $\mathcal{I}$ by the assignement $\{x_1 \leftarrow r_1, \ldots, x_n \leftarrow r_n\}$. For the sake of convenience we allow infix notion, for instance $(x)w(y)$ instead of $w(x, y)$. Furthermore, in the examples, we will sometimes use tuples of variables instead of a single variables. In this case of course we have to replace the corresponding quantifiers by quantifier strings of the same kind.

The set of formulas over a given basis is split up into fragments. According to [Rog87] the *number of quantor alternations* of a formula in prenex normal form ([Gal86]) is "the number of pairs of adjacent but unlike quantifiers". If this number is n and the outermost quantifier is $\exists$ (resp. $\forall$) the formula belongs to the Σ_{n+1}- (resp. Π_{n+1}-) fragment. $\Sigma_0 = \Pi_0$ denotes the set of quantifier-free formulas. An

abitrary formula belongs to a certain fragment if it is logically equivalent to a prenex normal form formula contained in this fragment.

Given a set Σ of symbols Σ^* denotes the set of finite and Σ^+ the set of finite nonempty strings over Σ. $\lhd$ is the prefix ordering on strings. A *Post Correspondence Problem* P over an alphabet Σ ([Pos46]) is given by a finite set of the form

$$\{(p_i, q_i) \mid 0 \leq i \leq m; p_i, q_i \in \Sigma^+\}$$

A *P-construction sequence* for $(u, v) \in \Sigma^* \times \Sigma^*$ is a sequence $\left((u_j, v_j)\right)_{j=1\ldots n}$ with

1. $u_j, v_j \in \Sigma^*$ for all j
2. $u_1 = v_1 = \epsilon$
3. $u_n = u$ and $v_n = v$
4. for each $1 \leq j \leq n-1$ there is a $0 \leq i \leq m$ with $u_{j+1} = u_j p_i$ and $v_{j+1} = v_j q_i$ where juxtaposition denotes the concatenation of strings.

In this case (u, v) is called *P-constructable*. P is *solvable* if there is an $u \in \Sigma^+$ such that (u, u) is P-constructable.

Equational problems emerged from the study of unification problems that can now be considered as a special case of equational problems (see [Sie89] for a survey on unification). For a set F of ranked function symbols let $T(F)$ denote the set of F-ground terms and $T(F, X)$ the set of F-terms that contain variables from the set X. $T(F)$ and $T(F, X)$ will also be considered as F-algebras where the symbols from F are given their Herbrand interpretation ([Gal86]). The basis and the model for equational problems are defined by an equational specification (F, E) in the sense of [EM85], here restricted to the one-sorted case, that is F is a ranked set of function symbols and E is a set of implicitly universally quantified equations of F-terms. The only predicate symbol is the equality symbol, the set of function symbols is given by the specification. [BS89] designate the following models of a specification (F, E):

- the *initial algebra* is the quotient of the ground term algebra $T(F)$ by the congruence generated by E.
- the *E-free algebra* is the quotient of the term algebra $T(F, X)$ by the congruence generated by E where X is a not further specified infinite set of variables.

A discussion of term algebras can be found in [EM85]. In this context [BS89] call the Π_3 fragment *special equational problems* and the Σ_2 fragment *special equational problems without independent parameters.*

The *lexicographic path ordering* on $T(F)$ has been described in [Der87][2] as a tool for proving termination of term rewriting systems. For a given partial order[3] $<_F$ on the set F of function symbols the lexicographic path ordering $\preceq_{\text{lpo}}$ is recursively defined by

$$t = g(t_1, \ldots, t_n) \preceq_{\text{lpo}} f(s_1, \ldots, s_m) = s$$

iff $t = s$ or one of the following holds

- $t \preceq_{\text{lpo}} s_i$ for some i
- $g <_F f$ and $t_j \prec_{\text{lpo}} s$ for all j
- $f = g$ and there is a $j \leq n$ with
  - $t_i = s_i$ for all $i < j$
  - $t_j \prec_{\text{lpo}} s_j$
  - $t_i \prec_{\text{lpo}} s$ for all $i > j$

where $x \prec_{\text{lpo}} y$ is an abbreviation for $x \preceq_{\text{lpo}} y \wedge x \neq y$. $\preceq_{\text{lpo}}$ is a simplification ordering ([Der87]), especially it is a partial order containing the subterm ordering. $\preceq_{\text{lpo}}$ is total iff the underlying precedence $<_F$ is total.

In the context of ordering relations we will also consider algebras containing finite *and* infinite trees. [Cou83] contains a treatement of infinite trees.

$f^n(t)$ means n applications of the unary function symbol f to the term t. $lth(s)$ denotes the length of the sequence s. $\square$ designates the end of a proof, the end of an example will be marked by $\diamond$.

3 Simulation of Strings

The first thing we need for the representation of the data type string is a coding function

$$\phi: \{a, b\}^* \to \mathcal{I}$$

We will use the symbol ϕ also to denote the corresponding function $\phi: \{a, b\}^* \times \{a, b\}^* \to \mathcal{I}^2$. The operations that will be used in the simulation of Post's Correspondence Problem are the test for emptiness and for each single nonempty string a unary function that appends this fixed string to its argument. For the sake of generality this function will be represented as a formula instead of a term. More precisely, we need:

- $\underline{\texttt{is-}\epsilon}(x)$

[2] referring to an unpublished paper of Kamin and Lévy.

[3] The definition in [Der87] (precedence) is slightly more general in using quasi-orderings.

- $(y)\underline{v}(x)$ for each $v \in \{a,b\}^+$

such that

[INJ] ϕ is injective

[EPS] For all $r \in \mathcal{I}$: $\mathcal{I} \models \underline{\texttt{is-}\epsilon}[r]$ iff $r = \phi(\epsilon)$

[CON] For all $r \in \mathcal{I}$, $v \in \{a,b\}^+$, $w \in \{a,b\}^*$: $\mathcal{I} \models [r]\underline{v}[\phi(w)]$ iff $r = \phi(wv)$

In applications we have to specify both ϕ and the formulas $\underline{\texttt{is-}\epsilon}$ and $\underline{v}$. This procedure contains a certain redundancy, an alternative is to give a different set of requirements on $\underline{\texttt{is-}\epsilon}$ and $\underline{v}$ such that the representation function can be derived from the definition of these formulas:

$$\begin{aligned} \phi(\epsilon) &:= \text{the unique } r \text{ with } \models \underline{\texttt{is-}\epsilon}[r] \\ \phi(w) &:= \text{the unique } r \text{ with } \models [r]\underline{w}[\phi(\epsilon)] \quad (w \neq \epsilon) \end{aligned}$$

We do not follow this line since it seems to be more natural to define the representation of strings explicitly. An advantage of this alternative way is that the requirements substituting [INJ], [EPS] and [CON] state only properties of the *theory of* the model instead of properties of the model itself. Anyway, with the next requirement we have no hope of staying within the scope of first order logic as has been explained in the introduction.

Definition 1 *$\sqsubset$ is the relation on $\mathcal{I}$ defined by: $x \sqsubset y$ iff there is a $v \in \{a,b\}^+$ with $\mathcal{I} \models [y]\underline{v}[x]$. As usual $\sqsubset^*$ denotes the reflexive transitive closure of $\sqsubset$. Furthermore $\sqsubset$ generalizes to pairs of objects by $(x_1,x_2) \sqsubset (y_1,y_2)$ iff $x_1 \sqsubset y_1$ and $x_2 \sqsubset y_2$.*

If $r_1 = \phi(w_1)$ and $r_2 = \phi(w_2)$ then $r_1 \sqsubset r_2$ expresses the prefix relationship between w_1 and w_2. However the definition is not restricted to representatives of strings, we will need this definition and the pertaining requirement in its full generality later. The formula $\underline{\texttt{finite}}$ characterizes the set of elements of the universe where $\sqsubset$ is a Noetherian relation. This set has to contain *at least* (but may not be equal to) the image of ϕ.

[NOE] There is no infinite descending $\sqsubset$-chain $(r_i)_{i\geq 0}$ in $\mathcal{I}$ with $\mathcal{I} \models \underline{\texttt{finite}}[r_0]$.

[FIN] For all $w \in \{a,b\}^+$: $\mathcal{I} \models \underline{\texttt{finite}}[\phi(w)]$

Example (1): The basis B contains at least the function symbols $\epsilon(0), a(1), b(1)$ and the equality symbol $=(2)$. Let $\mathcal{I}$ be the algebra of B-ground terms modulo some set of equations that do not involve any of the symbols ϵ, a, b.

Deliberately confusing the characters a, b from the alphabet with the unary function symbols a, b we define

$$\begin{aligned}
\phi(\sigma_0 \cdots \sigma_n) &:= \sigma_n(\cdots(\sigma_0(\epsilon))\cdots) \\
\underline{\texttt{is-}\epsilon}(x) &:= x = \epsilon \\
(y)\underline{\sigma_0 \cdots \sigma_n}(x) &:= y = \sigma_n(\cdots(\sigma_0(x))\cdots) \\
\underline{\texttt{finite}}(x) &:= \text{TRUE}
\end{aligned}$$

The reader might easily check that these definitions fulfill the requirements. ◇

Example (2): The basis B contains at least the function symbols $\epsilon(0), f(2)$ and the equality symbol $=(2)$. Analogously to Example (1) let $\mathcal{I}$ denote the algebra of B-ground terms modulo some set of equations that do not involve any of ϵ, f. With the following temporary definitions:

$$\begin{aligned}
\overline{a}(t) &:= f(\epsilon, t) \\
\overline{b}(t) &:= f(f(\epsilon, \epsilon), t)
\end{aligned}$$

we can define

$$\begin{aligned}
\phi(\sigma_0 \cdots \sigma_n) &:= \overline{\sigma_n}(\cdots(\overline{\sigma_0}(\epsilon))\cdots) \\
\underline{\texttt{is-}\epsilon}(x) &:= x = \epsilon \\
(y)\underline{\sigma_0 \cdots \sigma_n}(x) &:= y = \overline{\sigma_n}(\cdots(\overline{\sigma_0}(x))\cdots) \\
\underline{\texttt{finite}}(x) &:= \text{TRUE}
\end{aligned}$$

The above definitions still constitute a correct representation of strings when we enlarge the model $\mathcal{I}$ to the free algebra $T(F, X)$ modulo E. ◇

The next example shows a nontrivial $\underline{\texttt{finite}}$ formula.

Example (3): B contains at least the function symbols $\epsilon(0), a(1), b(1)$ and the predicate symbols $=(2), \leq(2)$. Consider the algebra $\mathcal{I}$ of finite and infinite B-ground terms where $\leq$ is interpreted as the subterm relation.[4] We choose $\phi, \underline{\texttt{is-}\epsilon}$ and $\underline{v}$ as in Example (1). The set of finite objects consists now of the terms built only with unary function symbols and containing the symbol ϵ.

$$\underline{\texttt{finite}}(x) := \epsilon \leq x \wedge \forall x'.x' \leq x \supset \left\{x' = \epsilon \vee \exists x''.x' = a(x'') \vee x' = b(x'')\right\}$$

If the set of nonunary function symbols $B' \subseteq B$ is finite we can transform the conclusion of the above implication into a Π_1-formula, thus saving one alternation of quantifiers:

$$\underline{\texttt{finite}}(x) := \epsilon \leq x \wedge \forall x'.x' \leq x \supset \bigwedge_{f \in B'} \forall \vec{z}.x' \neq f(\vec{z}) \qquad \diamond$$

[4]Note that the case of finite terms only is covered by Example (1).

4 Solutions of P

We are now ready to define the subformula $\underline{\texttt{one-step}}_P$. The intented meaning of $\underline{\texttt{one-step}}_P(y_1, y_2, y_3, y_4)$ is: "The pair of strings represented by (y_1, y_2) is obtained from the pair of strings represented by (y_3, y_4) by the application of one P-construction step." This is the only subformula that depends directly on the Post Correspondence Problem P:

$$\underline{\texttt{one-step}}_P(y_1, y_2, y_3, y_4) \quad := \quad \bigvee_{i=0,\ldots,m} ((y_1)\underline{p_i}(y_3) \wedge (y_2)\underline{q_i}(y_4))$$

where $P = \{(p_i, q_i) \mid i = 0, \ldots, m\}$.

4.1 Simulation of Sequences as Sets

In order to construct the sentence $\underline{\texttt{solvable}}_P$ we have to formulate something like "there is a P-construction sequence such that $\cdots$". How can we express as a formula the fact that something represents a P-construction sequence? The key idea we are going to explore now is: Instead of talking directly about sequences we may view a P-construction sequence as a *set* of pairs of strings. Since by definition a P-construction sequence is strictly ordered by the prefix relation on (pairs of) strings we are able to retain the sequence from the set.

With this idea we can now define the subformula $\underline{\texttt{constr}}_P(x)$ meaning that x represents a P-construction sequence. $\underline{\texttt{constr}}_P$ uses the subformula $(y_1, y_2)\underline{\texttt{in}}(x)$ reflecting the element relationship, the definition of which depends again on the model under consideration. From now on let a fixed Post Correspondence Problem P be given.

$$\begin{aligned}
\underline{\texttt{constr}}_P(x) := \forall y_1, y_2.(y_1, y_2)\underline{\texttt{in}}(x) \supset & \\
\{\underline{\texttt{is-}\epsilon}(y_1) \wedge \underline{\texttt{is-}\epsilon}(y_2)\} \vee & \qquad (2)\\
\exists y_3, y_4.(y_3, y_4)\underline{\texttt{in}}(x) \wedge \underline{\texttt{one-step}}_P(y_1, y_2, y_3, y_4) & \qquad (3)
\end{aligned}$$

Still leaving pending the definition of $\underline{\texttt{in}}$ we can now show

Lemma 1 *For all $r_1, r_2, u, s \in \mathcal{I}$ with $(r_1, r_2) \sqsubset^* (u, u)$ and*

$$\begin{aligned}
\mathcal{I} &\models \underline{\texttt{finite}}[u]\\
\mathcal{I} &\models \underline{\texttt{constr}}_P[s]\\
\mathcal{I} &\models [r_1, r_2]\underline{\texttt{in}}[s]
\end{aligned}$$

If [INJ], [EPS], [CON] and [NOE] are fulfilled then $(r_1, r_2) \in \textsc{Im}(\phi) \times \textsc{Im}(\phi)$ and the associated pair of strings $\phi^{-1}(r_1, r_2)$ is P-constructable.

Proof: We fix u and s with the above properties. Because of [NOE] there can not exist an infinite descending (w.r.t. $\sqsubset$) chain of pairs $(r_1, r_2) \sqsubset^* (u, u)$. We can therefore perform Noetherian induction on (r_1, r_2).

If $\mathcal{I} \models \underline{\texttt{is-}\epsilon}[r_1] \wedge \underline{\texttt{is-}\epsilon}[r_2]$ then [EPS] yields $(r_1, r_2) = \phi(\epsilon, \epsilon)$ and we are done.

Otherwise case (3) from the definition of $\underline{\texttt{constr}}_P$ applies, so there exist r_3, r_4 with $\mathcal{I} \models [r_3, r_4]\underline{\texttt{in}}[s]$ and $\mathcal{I} \models \underline{\texttt{one-step}}_P[r_1, r_2, r_3, r_4]$. From the definition of $\underline{\texttt{one-step}}_P$ follows $(r_3, r_4) \sqsubset (r_1, r_2) \sqsubset^* (u, u)$. The induction hypothesis yields that $(r_3, r_4) \in \text{IM}(\phi) \times \text{IM}(\phi)$ and $\phi^{-1}(r_3, r_4)$ is P-constructable, and because of [CON] and the definition of $\underline{\texttt{one-step}}_P$ the same holds for (r_1, r_2). □

We are now ready to define $\underline{\texttt{solvable}}_P$.

$$\underline{\texttt{solvable}}_P \quad := \quad \exists x, y.\underline{\texttt{constr}}_P(x) \wedge \underline{\texttt{finite}}(y) \wedge (y, y)\underline{\texttt{in}}(x) \wedge \neg\underline{\texttt{is-}\epsilon}(y)$$

From the above lemma we get immediately

Corollary 1 *If [INJ], [EPS], [CON] and [NOE] are fulfilled then*

$$\mathcal{I} \models \underline{\texttt{solvable}}_P \quad \Rightarrow \quad P \textit{ is solvable}$$

The reader should note that up to now we did not need any constraints on the subformula $\underline{\texttt{in}}$. We made use of the special properties of the model only in order to fulfill the requirements in connection with the simulation of strings. Once the representation of strings with the subformulas $\underline{\texttt{is-}\epsilon}, \underline{v}, \underline{\texttt{finite}}$ is found we get the first direction of our "goal"–theorem (1) for free — that is without worrying about the representation of sequences.

In order to prove the opposite direction of (1) we now have to choose a representation function for P-construction sequences and a corresponding formula $(y_1, y_2)\underline{\texttt{in}}(x)$. M denotes the domain of the representation function $\psi: M \to \mathcal{I}$:

$$M := \{(u_i, v_i)_{i=1\ldots n} \quad | \quad u_i, v_i \in \{a, b\}^*, n \geq 2, \\ u_i \lhd u_{i+1}, v_i \lhd v_{i+1}, (u_1, v_1) = (\epsilon, \epsilon)\}$$

Our last requirement relates the representation function ψ with the subformula $\underline{\texttt{in}}$ that is supposed to express the element relationship:

[IN] For all $s \in M$: $\mathcal{I} \models [r_1, r_2]\underline{\texttt{in}}[\psi(s)]$ iff there exits $j \in \{1, \ldots, lth(s)\}$ with $(r_1, r_2) = \phi(s(j))$

Lemma 2 *If [EPS], [CON], [FIN], [IN] are fulfilled then*

$$P \textit{ is solvable} \quad \Rightarrow \quad \mathcal{I} \models \underline{\texttt{solvable}}_P$$

The next theorem summarizes the method as it stands now:

Theorem 1 *Let B be a predicate logic basis and $\mathcal{I}$ a model for B. If we can find representation functions ϕ, ψ and formulas $\underline{\texttt{is-}\epsilon}, \underline{v}, \underline{\texttt{finite}}, \underline{\texttt{in}}$ such that [INJ], [EPS], [CON], [FIN], [NOE] and [IN] are fulfilled then the first order theory of $\mathcal{I}$ is undecidable.*

Now we can complete the examples started in Section 3:

Example (A): Consider equational problems for the equational specification $(F_A, AC(+))$ where $F_A := \langle \epsilon(0), a(1), b(2), f(2), +(2)\rangle$ and $AC(+)$ denotes the axioms of associativity and commutativity for $+$:

$$\begin{aligned} x + y &= y + x \\ (x + y) + z &= x + (y + z) \end{aligned}$$

We take the representation of strings from Example (1). It is easy to see that with the following definitions [IN] is fulfilled in the initial and in the free algebra:

$$\begin{aligned} \psi((u_i, v_i)_{i=1,\ldots,n}) &:= f(\phi(u_1), \phi(v_1)) + \cdots + f(\phi(u_n), \phi(v_n)) \\ (y_1, y_2)\underline{\texttt{in}}(x) &:= \exists x'.x = f(y_1, y_2) + x' \end{aligned}$$

Theorem 2 *The (Σ_3 fragment of the) first order theory of a ground term algebra (resp. term algebra) modulo associativity and commutativity is undecidable.*

We can improve this result by restricting the base to $F_{A'} := \langle \epsilon(0), f(2), +(2)\rangle$ and $P_{A'} := \langle =(2)\rangle$. With the representation of strings as in example (2) and $\psi, \underline{\texttt{in}}$ as above we obtain undecidability of the first order theory of $T(F_{A'})/_{AC(+)}$ and of $T(F_{A'}, X)/_{AC(+)}$. ◇

Example (B): Example (A) can be varied by enlarging the set of equations. If we add one further axiom for the idempotency of $+$:

$$x + x = x$$

we can use exactly the same setting to show

Theorem 3 *The (Σ_3 fragment of the) first order theory of a ground term algebra (resp. term algebra) modulo associativity,commutativity and idempotency is undecidable.* ◇

Example (C): Let $F_C := \langle \epsilon(0), a(1), b(1), e(1), l(1), h(3)\rangle$ and $P_C := \langle =(2), \leq(2)\rangle$. $\mathcal{I}_C$ is the ground term algebra $T(F_C)$ where $\leq$ is interpreted as the lexicographic path ordering $\preceq_{\text{lpo}}$ generated by the following precedence on F_C:

$$\epsilon <_F a <_F b <_F h <_F \begin{cases} l \\ e \end{cases}$$

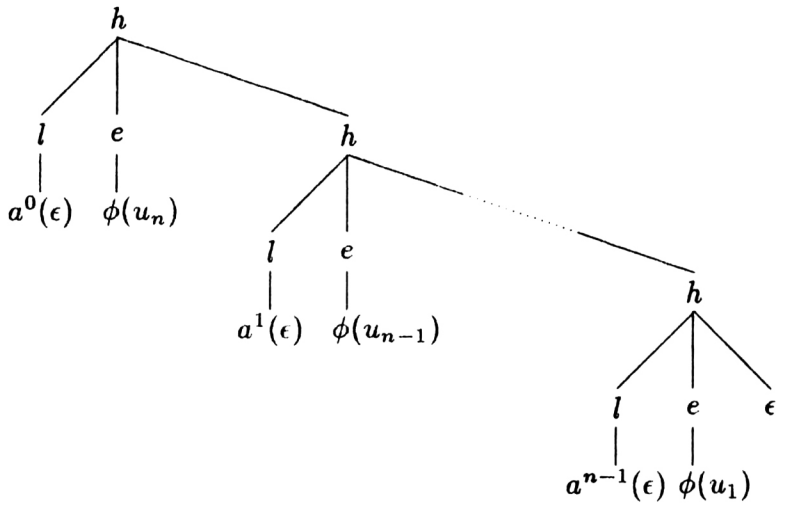

Figure 1: The term $\delta((u_i)_{i=1...n})$ representing the sequence $(u_i)_{i=1...n}$

e and l are uncomparable in the order $<_F$.

ϕ, $\underline{\texttt{is-}\epsilon}$, $\underline{\texttt{finite}}$ and $\underline{v}$ can be copied from example (1). A P construction sequence will be represented by two lists of labeleld strings, one for the first component and one for the second. The labels associate the corresponding components of a pair, moreover they will be essential for the formulation of the membership relation.

We associate to each nonempty sequence $s = (u_i)_{i=1,...,n}$ the term $\delta(s)$ as shown in Figure 1 and choose

$$\psi((u_i, v_i)_{i=1,...,n}) := (\delta((u_i)_{i=1,...,n}), \delta((v_i)_{i=1,...,n}))$$

In order to formulate the subformula $\underline{\texttt{in}}$ we use the following temporary definition:

$$(y)\underline{\texttt{in}}(x)\underline{\texttt{at}}(z) := h(l(z), e(y), \epsilon) \leq x \wedge \quad (4)$$

$$\forall y'.h(l(z), e(y'), \epsilon) \leq x \supset y' \leq y \quad (5)$$

Finally we define

$$(y_1, y_2)\underline{\texttt{in}}(x_1, x_2) := \exists z.(y_1)\underline{\texttt{in}}(x_1)\underline{\texttt{at}}(z) \wedge (y_2)\underline{\texttt{in}}(x_2)\underline{\texttt{at}}(z)$$

The proof of [IN] is given in Appendix (A.1).

Theorem 4 *The (Σ_4 fragment of the) first order theory of a partial lexicographic path ordering is undecidable.*

The separation of the P-construction sequence into two lists is not essential for the proof. In fact an analogous proof where the P-construction sequence is represented

by one list of pairs of strings is also possible (by changing the arity of h to 4). The price of this variant is the need for another maximal function symbol uncomparable to e and l, thereby leading to a "less total" ordering. In this alternative proof the labels $l(a^i(\epsilon))$ can not be omitted, they are necessary for the maximality condition in the definition of $\underline{\texttt{in}}$.

We remark that the same construction — with an appropriate modification of the technical lemma in Section A.1 — can be used to show that the Σ_4 fragment of the first order theory of a recursive path ordering([Der82]) is undecidable. $\diamond$

Counting the quantifiers involved in the above construction we find that the formula $\underline{\texttt{solvable}}_P$ is at least in the Σ_3 fragment. This is an inherent drawback of this method since $\underline{\texttt{solvable}}_P$ follows the pattern

$$\exists s \cdots \forall (s_1, s_2) \in s \cdots \exists (s_3, s_4) \in s \cdots$$

In general the formula $\underline{\texttt{in}}$ is the most "expensive" one (in terms of alternations of quantifiers). We will always try to find a formula $\underline{\texttt{in}}$ in Σ_1, if we do not succeed we get undecidability only for a fragment larger than Σ_3.

4.2 Direct Simulation of Sequences

In some applications it is possible to overcome this limitation by using a direct simulation technique for sequences. In this case we have to perform three different operations on the data type sequence, and we have to work a little bit harder to regulate the correlation of the pertaining formulas. We will come back to a comparison of these two methods at the end of this section.

The formulas that are to be designed for the model under consideration are

- $\underline{\texttt{nonempty}}(x)$
- $(y_1, y_2, x')\underline{\texttt{sub-of}}(x)$
- $(y_1, y_2)\underline{\texttt{head-of}}(x)$

The intended meaning of the first formula should be clear, $(y_1, y_2, x')\underline{\texttt{sub-of}}(x)$ is supposed to express that the sequence with first element (y_1, y_2) and tail x' is a suffix of the sequence x, and $(y_1, y_2)\underline{\texttt{head-of}}(x)$ is intended to express that (y_1, y_2) is the head of the sequence x.

The analogous definition of $\underline{\texttt{constr}}_P$ is now

$$\begin{aligned}\underline{\texttt{constr}}_P(x) := \forall y_1, y_2, x'.(y_1, y_2, x')\underline{\texttt{sub-of}}(x) \supset \\ \{\underline{\texttt{is-}\epsilon}(y_1) \wedge \underline{\texttt{is-}\epsilon}(y_2)\} \vee \\ \{\underline{\texttt{nonempty}}(x') \wedge \forall y_3, y_4.(y_3, y_4)\underline{\texttt{head-of}}(x') \supset \underline{\texttt{one-step}}_P(y_1, y_2, y_3, y_4)\}\end{aligned}$$

and finally the formula $\underline{\texttt{solvable}}_P$ reads

$$\underline{\texttt{solvable}}_P := \exists x, y.\underline{\texttt{constr}}_P(x) \wedge (y, y)\underline{\texttt{head-of}}(x) \wedge \underline{\texttt{finite}}(y) \wedge \neg\underline{\texttt{is-}\epsilon}(y)$$

In contrast to section 4.1 where we obtained the first direction of (1) just from the representation of strings we now have to state additional requirements on the newly introduced formulas:

[NH] $\mathcal{I} \models \forall x.\underline{\texttt{nonempty}}(x) \supset \exists y_1, y_2.(y_1, y_2)\underline{\texttt{head-of}}(x)$

[HS] $\mathcal{I} \models \forall x, y_1, y_2.(y_1, y_2)\underline{\texttt{head-of}}(x) \supset \exists x'.(y_1, y_2, x')\underline{\texttt{sub-of}}(x)$

[HSH] $\mathcal{I} \models \forall x, x', y_1, y_2, y_3, y_4.(y_1, y_2, x')\underline{\texttt{sub-of}}(x) \wedge (y_3, y_4)\underline{\texttt{head-of}}(x') \supset \exists x''.(y_3, y_4, x'')\underline{\texttt{sub-of}}(x)$

At this point the reader might remark that we could have used [NH] as definition of nonempty by turning the implication sign into an equivalence. In this case only the requirement [HS] and [HSH] remain relating head-of to sub-of. We do not choose this approach in order to avoid the introduction of extra quantifiers. Example (D) shows how a model specific argument leads to the elimination of an unwanted existential quantifier in the definition of nonempty.

With the help of these properties we can now prove a lemma analogous to Lemma 1:

Lemma 3 *For all* $r_1, r_2, u, s, s' \in \mathcal{I}$ *with* $(r_1, r_2) \sqsubset^* (u, u)$ *and*

$$\mathcal{I} \models \underline{\texttt{finite}}[u]$$
$$\mathcal{I} \models \underline{\texttt{constr}}_P[s]$$
$$\mathcal{I} \models [r_1, r_2, s']\underline{\texttt{sub-of}}[s]$$

If [INJ], [EPS], [CON], [NOE], [NH] and [HSH] are fulfilled then $(r_1, r_2) \in \text{IM}(\phi) \times \text{IM}(\phi)$ *and* $\phi^{-1}(r_1, r_2)$ *is P-constructable.*

Proof: As in the proof of Lemma 1 we proceed by Noetherian induction on (r_1, r_2).

If $\mathcal{I} \models \underline{\texttt{is-}\epsilon}[r_1] \wedge \underline{\texttt{is-}\epsilon}[r_2]$ we know from [EPS] that $(r_1, r_2) = \phi(\epsilon, \epsilon)$.

Otherwise $\mathcal{I} \models \underline{\texttt{nonempty}}[s]$, so we get from [NH] that there are $r_3, r_4 \in \mathcal{I}$ with $\mathcal{I} \models [r_3, r_4]\underline{\texttt{head-of}}[s]$. The second case from the definition of $\underline{\texttt{constr}}_P$ applies and we get $\mathcal{I} \models \underline{\texttt{one-step}}_P[r_1, r_2, r_3, r_4]$, this implies $(r_3, r_4) \sqsubset (r_1, r_2) \sqsubset^* (u, u)$. Because of [HSH] there is a $s'' \in \mathcal{I}$ with $\mathcal{I} \models [r_3, r_4, s'']\underline{\texttt{sub-of}}[s]$, so we can apply the induction hypothesis to (r_3, r_4). With [CON] and the definition of $\underline{\texttt{one-step}}_P$ the proof is completed. □

Corollary 2 *If [INJ], [EPS], [CON], [NOE], [NH], [HSH] and [HS] are fulfilled then*

$$\mathcal{I} \models \underline{\texttt{solvable}}_P \quad \Rightarrow \quad P \textit{ is solvable}$$

In a first attempt we could require as in Section 4.1 a coding function mapping the set M into $\mathcal{I}$. This will suffice in some examples, but we can be more liberal and allow for each sequence s a "private" coding function for the set of the subsequences of s:

$$\psi \in \prod_{s \in M} (\{0, \ldots, lth(s)\} \to \mathcal{I})$$

The subformulas nonempty, sub-of and head-of have to work properly for the codings of subsequences:

For all $s \in M, n \leq lth(s)$:

[NIL] $\mathcal{I} \models \underline{\texttt{nonempty}}[\psi(s)(n)]$ iff $n \neq 0$

[HEA] $\mathcal{I} \models [r_1, r_2]\underline{\texttt{head-of}}[\psi(s)(n)]$ iff $n \geq 1$ and $(r_1, r_2) = \phi(s(n))$

[SUB] $\mathcal{I} \models [r_1, r_2, t]\underline{\texttt{sub-of}}[\psi(s)(lth(s))]$ iff there is $i \in \{1, \ldots, lth(s)\}$ with $(r_1, r_2) = \phi(s(i))$ and $t = \psi(s)(i-1)$

Lemma 4 *If [EPS], [CON], [FIN], [NIL], [HEA] and [SUB] are fulfilled then*

$$P \text{ is solvable} \quad \Rightarrow \quad \mathcal{I} \models \underline{\texttt{solvable}}_P$$

Theorem 5 gives the complete method developed in this section:

Theorem 5 *Let B be a predicate logic basis and $\mathcal{I}$ a model for B. If we can find representations ϕ, ψ and formulas* $\underline{\texttt{is-}\epsilon}$, $\underline{v}$, $\underline{\texttt{finite}}$, $\underline{\texttt{head-of}}$ *and* $\underline{\texttt{sub-of}}$ *such that [INJ], [EPS], [CON], [NOE], [NH], [HS], [HSH], [FIN], [NIL], [HEA] and [SUB] are fulfilled, then the first order theory of $\mathcal{I}$ is undecidable.*

Example (D): Let us now see how the undecidability result for the theory of subterm ordering from [Ven87] fits into our framework:

Let $F_D := \langle \epsilon(0), a(1), b(1), f(3) \rangle$ and $P_D := \langle =(2), \leq(2) \rangle$. $\mathcal{I}_D$ is the algebra of F_D-ground terms where $\leq$ is interpreted as the subterm relation. The representation of strings has been given in Example (1). We choose $\psi(s)(i)$ as shown in Figure 2 if $i \geq 1$, $\psi(s)(0) = \epsilon$, and define the remaining formulas:

$$\begin{aligned} (y_1, y_2)\underline{\texttt{head-of}}(x) &:= \exists x'.x = f(y_1, y_2, x') \\ (y_1, y_2, x')\underline{\texttt{sub-of}}(x) &:= f(y_1, y_2, x') \leq x \\ \underline{\texttt{nonempty}}(x) &:= \exists y_1, y_2, x'.x = f(y_1, y_2, x') \end{aligned}$$

We can save one alternation of quantifiers in $\underline{\texttt{solvable}}_P$ by transforming nonempty into a Π_1 formula[5].

$$\underline{\texttt{nonempty}}(x) \quad := \quad x \neq \epsilon \wedge \forall x'.x \neq a(x') \wedge x \neq b(x')$$

[5] In ground term algebras over a finite alphabet it is always possible to transform a purely equational formula into a Π_1 (or Σ_1) formula, see [CL89].

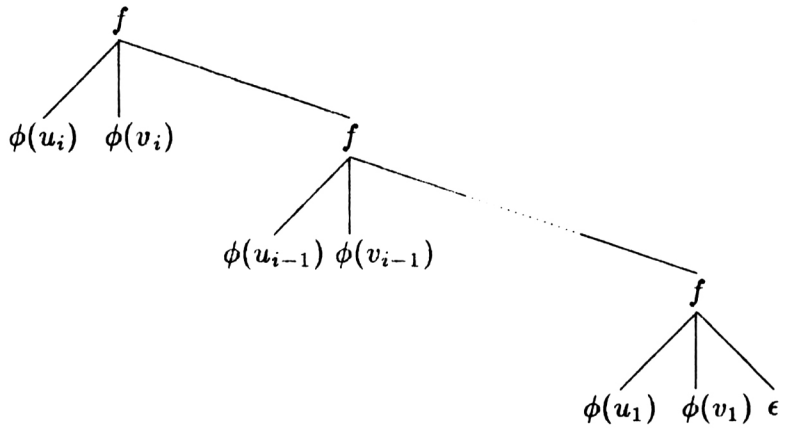

Figure 2: The term $\psi((u_j, v_j)_{j=1...n})(i)$ representing the subsequence $(u_j, v_j)_{j=1...i}$ of the sequence $(u_j, v_j)_{j=1...n}$ for $1 \leq i \leq n$

Theorem 6 ([Ven87]) *The (Σ_2-fragment of the) first order theory of the subterm ordering is undecidable.* ◇

Example (E): We can modify the above example by enlarging the model to the algebra of finite and infinite ground terms. We can use exactly the same proof as above but with the **finite** formula as in Example (3) to show

Theorem 7 *The (Σ_2 fragment of the) first order theory of the subterm ordering in the algebra of finite and infinite trees in undecidable.* ◇

Example (F): If we drop commutativity from example (A) we can now show undecidability even of the Σ_2 fragment:

Consider the equational specification $(F_F, A(+))$ where $F_F = \langle \epsilon(0), f(2), +(2) \rangle$ and $A(+)$ denotes the axiom of associativity for $+$:

$$(x + y) + z \quad = \quad x + (y + z)$$

For the initial algebra we take the representation of strings from Example (2) and ψ similar to Example (A):

$$\begin{aligned}
\psi((u_i, v_i)_{i=1,\ldots,m})(j) &:= f(\phi(u_j), \phi(v_j)) + \cdots + f(\phi(u_1), \phi(v_1)) + \epsilon \quad (j \geq 1) \\
\psi((u_i, v_i)_{i=1,\ldots,m})(0) &:= \epsilon \\
(y_1, y_2)\underline{\texttt{head-of}}(x) &:= \exists x'.x = f(y_1, y_2) + x' \\
(y_1, y_2, x')\underline{\texttt{sub-of}}(x) &:= x = f(y_1, y_2) + x' \vee \exists x''.x = x'' + f(y_1, y_2) + x' \\
\underline{\texttt{nonempty}}(x) &:= x \neq \epsilon \wedge \forall y_1, y_2.x \neq f(y_1, y_2) \wedge x \neq \epsilon + y_1
\end{aligned}$$

Here the formula nonempty is again obtained by a quantifier elimination analogous to Example (D).

Theorem 8 *The (Σ_2 fragment of the) theory of a ground term algebra modulo associativity is undecidable.* ◇

In the examples (D) to (F) we gave uniform codings for the sequences. The last example (G) shows the use of "private" coding functions for the subsequences of a given sequence. As mentioned in the introduction this is an artificial example that serves just for the purpose of demonstrating the usage of our method in its full generality.

Example (G): Let $F_G := \langle 0(0), 1(0), +(2), *(2)\rangle$ and $P_G := \langle =(2), \leq(2)\rangle$. Our interpretation $\mathcal{I}_G$ is the model of natural numbers. In order to define the representation of strings we introduce two abreviations:

$$\begin{aligned} \overline{a}(t) &:= t + t \\ \overline{b}(t) &:= t + t + 1 \end{aligned}$$

It is easy to see that the following representation of strings fulfills the requirements since there is an obvious correspondence between strings and the binary representation of natural numbers.

$$\begin{aligned} \phi(\sigma_0 \cdots \sigma_n) &:= \overline{\sigma_n}(\cdots \overline{\sigma_0}(1)) \cdots) \\ \underline{\texttt{is-}\epsilon}(x) &:= x = 1 \\ (y)\underline{\sigma_0 \cdots \sigma_n}(x) &:= y = \overline{\sigma_n}(\cdots \overline{\sigma_0}(x)) \cdots) \\ \underline{\texttt{finite}}(x) &:= \textsc{True} \end{aligned}$$

We use Gödel's β-predicate to represent sequences in the domain of natural numbers. The existence of the representation ψ is a consequence of the fundamental property of the β-predicate. The definition of β and the pertaining theorem are restated in appendix A.2.

$$\begin{aligned} \underline{\texttt{nonempty}}(c, d, n) &:= n \geq 1 \\ (y_1, y_2)\underline{\texttt{head-of}}(c, d, n) &:= \beta(c, d, n + n, y_1) \wedge \beta(c, d, n + n + 1, y_2) \\ (y_1, y_2, (c', d', n'))\underline{\texttt{sub-of}}(c, d, n) &:= c' = c \wedge d' = d \wedge n' \leq n \\ &\quad \wedge (y_1, y_2)\underline{\texttt{head-of}}(c', d', n') \end{aligned}$$

As a result we obtain the undecidability of the Σ_2 fragment of complete number theory. The reader should note that $\textsc{Im}(\phi) = \mathcal{I} \setminus \{0\}$ — especially the images of ϕ and ψ are not disjoint. ◇

In this section we have finished the presentation of the two methods for proving the undecidability of the first order theory of a model. The first method is appropriate

for models that miss a concept of ordering (for instance term algebras modulo associativity and commutativity), while the second is applicable to models where some kind of ordering is present. In view of the fact that the second method can yield undecidability of a more simple fragment than the first one, the question arises why we did not use the second method for proving undecidability of the theory of a partial recursive path ordering in order to find a formula $\underline{\texttt{solvable}}_P$ in a smaller fragment than Σ_4.

The reason is that we can benefit from the simpler quantification structure of $\underline{\texttt{solvable}}_P$ in the second method only if we succeed in finding *simple* formulas $\underline{\texttt{nonempty}}$, $\underline{\texttt{sub-of}}$ and $\underline{\texttt{head-of}}$ fulfilling the requirements. More precisely, we get a formula $\underline{\texttt{solvable}}_P$ in Σ_3 iff $\underline{\texttt{nonempty}}$ is in $\Pi_2 \cup \Sigma_1$ and both $\underline{\texttt{sub-of}}$ and $\underline{\texttt{head-of}}$ are in $\Pi_1 \cup \Sigma_2$, provided that $\underline{\texttt{is-}\epsilon}$, $\underline{v}$ and $\underline{\texttt{finite}}$ do not induce any further alternation of quantifiers. This is usually the case, in all applications we found the expensive operations belong to the datatype set, resp. sequence. Using representations of sequences in the spirit of Example (C) one could try to define $\underline{\texttt{sub-of}}$ with the help of a maximality condition as it has been done in the definition of $\underline{\texttt{in}}$, but we did not succeed in finding a formula $\underline{\texttt{sub-of}} \in \Pi_1 \cup \Sigma_2$ fullfilling [HSH].

5 Conclusions

We have presented two methods for proving the undecidability of the first order theory of a model. In order to apply one of these methods to a given model we have to find appropriate representations of the data types "string" and "sequence" and formulas expressing the operations on these data types. The two main theorems (Theorem 1 and Theorem 5) state that the proof of undecidability is completed if the pertaining set of requirements is fulfilled. We would like to point out some statements that at a first glance one might expect to be essential for a reduction proof but that in fact are not. With the presentation of this list we claim that applications benefit from a systematic study of reduction proofs since it localizes the crucial points where the special properties of a model are involved.

- A general binary concatenation operation is not necessary. The reader might try to find such a formula in the case of representation of strings by unary function symbols (Example (1)).
- The codings of strings and sequences may be not disjoint. This has been used in Example (G).
- Formulas characterizing the images of the representations ϕ and ψ are not needed. In particular, it is not necessary to express that the elements of some set (sequence) are indeed pairs of strings.

- One should not worry about an *explicit* characterization of the finiteness of sequences in terms of first order logic.

In the undecidability proof of [Ven87] there exist subformulas in his construction that *explicitly* specify the shape of the objects that are intended to express P-construction sequences. In the approach presented here this is not necessary, we therefore yield a simpler formula $\underline{\texttt{solvable}}_P$.

There is a potentially useful extension to the method that was not carried out since there are no applications at hand. Strings have been coded in the universe of the model by a representation *function*, instead we could associate an equivalence class of the universe to each string. This implies the need for further restrictions that guarantee the congruence property of the operations on strings.

The starting point of the method proposed is the undecidability of Post's Correspondence Problem. Of course there are many other undecidable problems that might serve for reduction to the decision problem of a theory (see [Dav77]). One may, for instance, take the uniform halting problem for Turing machines and perform a reduction proof in the above style: A Turing machine halts iff there is a finite sequence of configurations such that the first and the last one are in some special form and such that each adjacent pair is related by some "local transformation". A configuration can be interpreted as a pair of strings (the part of the tape to the left, resp. to the right of the head). Hence the data types involved here are the same. Our choice of Post's Correspondence Problem is somewhat arbitrary but leads to a technically simpler proof.

Another popular candidate for reduction is complete number theory. One may use the result of [Mat70] on the unsolvability of Hilberts Tenth Problem and reduce the Σ_1-fragment hoping to obtain a formula $\underline{\texttt{solvable}}_P$ in a pretty small fragment. In fact [Qui46] gives a reduction of complete number theory to the theory of concatenation of strings over the alphabet $\{a, b\}$. The number n is coded by the string consisting of n a's, such that addition of numbers corresponds to the concatenation of strings. Multiplication is expressed with the help of lists that can be viewed as computation sequences for an iterative version of the multiplication algorithm. So it seems that this approach yields equally small fragments as ours, but the special point in Quine's proof is that a general concatenation operation is available in the logic, such that no quantifiers are needed for expressing addition. In most of the applications presented here we have just some kind of successor function given, in this case we need a list construction for expressing the addition operation. In order to optimize the alternations of quantifiers the iterative processes of addition and multiplication has to be performed in one list apparatus, thus yielding a more complex reduction proof.

In the introduction we mentioned some decidability results related to our applications, but there are still some gaps between these results and ours. We conclude with some open problems:

- The decidability of the Σ_1-fragment of the theory of ground term algebra modulo associativity and commutativity has been proved in [Com88]. While we have shown the undecidability of the Σ_3-fragment the Σ_2-case is still unsolved. Furthermore we may consider special sets of function symbols. [Com88] remarks that the case of one AC function symbol and one constant is equivalent to Presburger arithmetic and therefore decidable. Unsolved cases are one AC function symbol plus a finite set of constants (called the "theory of finitely generated multisets" in [Com90a]) and the case of one AC function symbol, one unary function symbol plus one constant.
- [Com90b] shows the decidability of the Σ_1-fragment of a total lexicographic path ordering, but the same question for the partial case remains open. On the other hand we have shown the undecidability of the Σ_4-fragment of the theory of *partial* lexicographic path ordering. We gave the proof for a precedence that is "as total as possible" but did not succeed in applying the technique to the total case. The reason is that at least two uncomparable function symbols are needed in order to distinguish between the two components of a pair. A proof of undecidability in the style presented here seems only to be possible beyond a purely symbolic level of representation, as illustrated in Example (G).

I am grateful to Jacques Loeckx and Stephan Uhrig for comments on a draft of this paper.

References

[BHS86] Hans-Jürgen Bürckert, Alexander Herold, and Manfred Schmidt-Schauß. *On Equational Theories, Unification and Decidability.* SEKI-Report SR–86–20, Universität Kaiserslautern, 1986.

[BS89] Hans-Jürgen Bürckert and Manfred Schmidt-Schauß. *On the Solvability of Equational Problems.* SEKI Report SR-89-07, Universität Kaiserslautern, 1989.

[CK73] C. C. Chang and H. J. Keisler. *Model Theory. Studies in Logic and the Foundations of Mathematics, vol. 73*, North-Holland Publishing Company, 1973.

[CL89] Hubert Comon and Pierre Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7(3,4):371–425, 1989.

[Com88] Hubert Comon. *Unification et Disunification. Théorie et Applications.* PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1988.

[Com90a] Hubert Comon. *Disunification:A Survey.* Rapport de Recherche no. 540, LRI, Université de Pairs Sud, january 1990.

[Com90b] Hubert Comon. Solving inequations in term algebras. In *5th Symposium on Logic in Computer Science*, 1990.

[Cou83] Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25(2):95–169, 1983.

[Dav77] Martin Davis. Unsolvable problems. In Jon Barwise, editor, *Handbook of Mathematical Logic*, chapter C.2, pages 567–594, North-Holland, 1977.

[Der82] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 7:279–301, 1982.

[Der87] Nachum Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3:69–116, 1987.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification, vol. 1. EATCS-Monographs on Theoretical Computer Science*, Springer-Verlag, 1985.

[End72] Herbert B. Enderton. *Mathematical Introduction to Logic.* Academic Press, 1972.

[Gal86] Jean H. Gallier. *Logic for Computer Science.* Harper & Row, publishers, 1986.

[God31] Kurt Gödel. Über Formal Unentscheidbare Sätze der Pricipia Mathematica und Verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931. Reprinted in [God86].

[God86] Kurt Gödel. *Collected Works.* Oxford University Press, 1986. 2 volumes.

[Kei71] H. Jerome Keisler. *Model Theory for Infinitary Logic. Studies in Logic and the Foundations of Mathematics, vol. 62*, North-Holland Publishing Company, 1971.

[Kir85] Claude Kirchner. *Méthodes et Outils de Conception Systématique d'Algorithmes d'Unification dans les Théories Équationelles.* PhD thesis, Centre de Récherche en Informatique de Nancy, 1985.

[Mat70] Yu Matijacevič. Enumerable sets are diophantine. *Dokl. Akad. Nauk. SSSR*, 191:279–282, 1970.

[Plo72] G. D. Plotkin. Building-in equational theories. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence 7*, pages 73–90, Edinburgh University Press, 1972.

[Pos46] Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the AMS*, 52:264–268, 1946.

[Qui46] W. V. Quine. Concatenation as a basis for arithmetic. *Journal of Symbolic Logic*, 11(4):105–114, 1946.

[Rab65] M. O. Rabin. A simple method for undecidability proofs and some applications. In Yehoshua Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science*, pages 58–68, North-Holland, 1965.

[Rog87] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* MIT Press, second edition, 1987.

[Sie89] Jörg H. Siekmann. Unification theory. A survey. *Journal of Symbolic Logic*, 1989.

[Sti81] Mark E. Stickel. A unification algorithm for associative-commutative functions. *Journal of the ACM*, 28(3):423–434, 1981.

[Tar53] Alfred Tarski. A general method in proofs of undecidability. In Alfred Tarski, Andrzej Mostowski, and Raphael M. Robinson, editors, *Undecidable Theories*, pages 1–35, North-Holland, 1953.

[TMR53] Alfred Tarski, Andrzej Mostowski, and Raphael M. Robinson. Undecidability and essential undecidability in arithmetic. In Alfred Tarski, Andrzej Mostowski, and Raphael M. Robinson, editors, *Undecidable Theories*, pages 37–74, North-Holland, 1953.

[Ven87] K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebra. *Journal of the ACM*, 34(2):492–510, april 1987.

A Appendix

A.1 Proof of [IN] for Example (C)

In the exact definition of δ we have to generalize to sequences starting with an arbitrary index $k \geq 1$:

$$\delta((u_i)_{i=k\ldots n}) = \begin{cases} \epsilon & \text{if } k > n \\ h(l(a^{k-1}(\epsilon)), e(\phi(u_n)), \delta((u_{i-1})_{i=k+1\ldots n})) & \text{otherwise} \end{cases}$$

In order to prove [IN] we need the following lemma:

Lemma 5 *Let $s = (u_i)_{i=1\ldots n}$ be a nonempty increasing sequence over $\{a,b\}^*$. Then for all $t, t_0 \in T(\{\epsilon,a,b\})$ the following two statements are equivalent:*

1. $\mathcal{I} \models [t]\underline{\mathbf{in}}[\delta(s)]\underline{\mathbf{at}}[t_0]$
2. *there exists $j \in \{1,\ldots,n\}$ with $t_0 = a^{n-j}(\epsilon)$ and $t = u_j$*

Proof: First we state a simple fact about the lexicographic ordering $\preceq_{\mathrm{lpo}}$ generated by an ordering $\leq_F$ on F:

(*) If $t_1 \preceq_{\mathrm{lpo}} t_2$ then for each operator symbol f occurring in t_1 there is a symbol g in t_2 such that $f \leq_F g$.

Now let $h(l(t_0),e(t),\epsilon) \preceq_{\mathrm{lpo}} \delta((u_i)_{i=k\ldots n})$. According to the definition of an lpo there are four possibilities:

1. $h(l(t_0),e(t),\epsilon) \preceq_{\mathrm{lpo}} l(a^{k-1}(\epsilon))$ or $h(l(t_0),e(t),\epsilon) \preceq_{\mathrm{lpo}} e(u_n)$
2. $h(l(t_0),e(t),\epsilon) \preceq_{\mathrm{lpo}} \delta((u_{i-1})_{i=k+1\ldots n})$
3. $l(t_0) = l(a^k(\epsilon))$ and $e(t) \preceq_{\mathrm{lpo}} e(u_n)$
4. $l(t_0) \prec_{\mathrm{lpo}} l(a^k(\epsilon))$ and $e(t) \prec_{\mathrm{lpo}} \delta((u_i)_{i=k\ldots n})$

Because of (*) possibility (1) can be dropped immediately. For the same reason (4) is only possible if $e(t) \preceq_{\mathrm{lpo}} e(u_j)$ for some $j \in \{k,\ldots,n\}$. With an inductive argument in case (2) and applying again (*) we get especially for $k = 1$:

(**) If $h(l(t_0),e(t),\epsilon) \preceq_{\mathrm{lpo}} \delta((u_i)_{i=1\ldots n})$ then there are i,i' with $1 \leq i' \leq i \leq n$ such that $t_0 \preceq_{\mathrm{lpo}} a^{n-i}(\epsilon)$ and $t \preceq_{\mathrm{lpo}} u_{i'}$.

(1) ⇒ (2): Let $\mathcal{I} \models (t)\underline{\mathbf{in}}(\delta(s))\underline{\mathbf{at}}(t_0)$. From (**) we know that there exist i,i' with

$$\begin{aligned} t_0 &\preceq_{\mathrm{lpo}} a^{n-i}(\epsilon) \\ t &\preceq_{\mathrm{lpo}} u_{i'} \end{aligned}$$

and $1 \leq i' \leq i \leq n$. Since there is no nonconstant smaller than a, t_0 must be of the form $a^{n-j}(\epsilon)$ with $i \leq j \leq n$. We therefore conclude

$$t \preceq_{\mathrm{lpo}} u_{i'} \preceq_{\mathrm{lpo}} u_i \preceq_{\mathrm{lpo}} u_j$$

On the other hand we know from the construction of $\delta(s)$ that

$$h(l(a^{n-j}(\epsilon)),e(u_j),\epsilon) \preceq_{\mathrm{lpo}} \delta(s)$$

and (5) yields $u_j \preceq_{\text{lpo}} t$. From the antisymmetry of $\preceq_{\text{lpo}}$ we get $t = u_j$.

(2) ⇒ (1): (4) follows immediately from the definition of $\delta(s)$. So let

$$h(l(a^{m-j}(\epsilon)), e(t'), \epsilon) \preceq_{\text{lpo}} \delta(s)$$

Because of (**) there are i, i' with $1 \leq i' \leq i \leq n$ and

$$\begin{array}{rcl} a^{m-j}(\epsilon) & \preceq_{\text{lpo}} & a^{m-i}(\epsilon) \\ t' & \preceq_{\text{lpo}} & u_{i'} \end{array}$$

This is only possible if $j \geq i$ and we obtain

$$t' \preceq_{\text{lpo}} u'_i \preceq_{\text{lpo}} u_i \preceq_{\text{lpo}} u_j = t$$

□

A.2 Gödels β-Predicate

The β predicate was introduced in [God31]. A proof of the theorem we restate below can also be found in textbooks on mathematical logic, for instance [End72].

$$\beta(x_1, x_2, l, x) := \exists q. x_1 = q * (1 + (l+1) * x_2) + x \wedge x < 1 + (l+1) * x_2$$

Theorem 9 ([God31]) *For each sequence $a_0, \ldots, a_n$ of natural numbers there exist c, d such that for all $i \leq n$:*

$$\mathcal{I}_G \models \beta[c, d, i, x] \quad \Leftrightarrow \quad x = a_i$$

We can now choose the representation ψ of Example (G):

$$\psi((u_i, v_i)_{i=1,\ldots,m})(j) \quad := \quad (c, d, 2 * j + 1)$$

where c, d are the values corresponding to the sequence

$$(0, 0, \phi(u_1), \phi(v_1), \ldots, \phi(u_m), \phi(v_m))$$