A Function Elimination Method for Checking Satisfiability of Arithmetical Logics

Valentina Castiglioni, Ruggero Lanotte, and Simone Tini

Dipartimento di Scienza e Alta Tecnologia, Università dell'Insubria

Abstract. We study function elimination for Arithmetical Logics. We propose a method allowing substitution of functions appearing in a given formula with functions with less arity. We prove the correctness of the method and we use it to show the decidability of the satisfiability problem for two classes of formulae allowing linear and polynomial terms.

1 Introduction

A branch of First Order Logics are Arithmetical Logics. Arithmetical Logics consider quantified formulae where variables are either reals or integers, and where functions in $\{+, \cdot\}$ and relations in $\{<, =, >, \leq, \geq\}$ are used.

The satisfiability problem deals with the existence of a valuation for variables, functions and predicates s.t. a given formula is true under that valuation. This problem is in general undecidable if one considers polynomial formulae on integer variables (see [23]). Different classes are considered to avoid such difficulty, and several techniques can be used to prove decidability (see [4]). One of these techniques is based on the quantifier elimination approach. Given a formula with a quantified variable, the quantifier elimination deals with the problem of finding an equivalent formula without quantifiers.

The results of this paper are the following ones: we prove that quantifier elimination for formulae with functions cannot exist; we introduce a technique that permits to decrease the arity of a certain function; we use the technique proposed to prove the decidability of the satisfiability problem for two subclasses of formulae in the set of formulae with prefixes in $(\exists^*\forall)^*$. The two classes (the first one allowing real polynomial terms and the second one mixed linear terms) are sufficient to describe the theories of sets and strings.

The paper is organized as follows: in Section 2 we review some base notions and results on Arithmetical Logics. In Section 3 we show the function elimination method and the two decidable classes are defined in Section 4. In Section 5 we use our method to model sets and in Section 6 we discuss related works and draw some conclusions.

2 Background

Let us assume a set of *function symbols* F ranged over by $a, b, \ldots$ together with an *arity* mapping $ar : F \to \mathbb{N}$. If $ar(a) = 0$ then a is called a *real variable*. We will usually use $x, y, \ldots$ to denote real variables.

A *valuation* over F is a mapping $v : F \to (\mathbb{R}^{\mathbb{N}} \to \mathbb{R})$ that assigns a function $v(a) : \mathbb{R}^{ar(a)} \to \mathbb{R}$ to each function symbol a. Notice that $v(x)$ is a constant for each variable x. The set of all valuations over F is denoted by $\mathcal{V}$. For a valuation $v \in \mathcal{V}$, a function symbol a and a function $f : \mathbb{R}^{ar(a)} \to \mathbb{R}$, we denote with $v[f/a]$ the valuation s.t. $v[f/a](a) = f$ and $v[f/a](b) = v(b)$ for all $b \neq a$.

Definition 1. *The set $\mathcal{T}$ of* polynomial terms with rational coefficients*, ranged over by τ, is defined inductively by:*

$$\tau ::= q \mid q \cdot a(\tau_1, \ldots, \tau_{ar(a)}) \mid \tau_1 + \tau_2 \mid \tau_1 \cdot \tau_2$$

where $q \in \mathbb{Q}$ and $a \in F$. The terms in $\mathcal{T}$ generated without using the multiplication $\tau_1 \cdot \tau_2$ are called linear terms*.*

When we use real variables in terms, we usually write x for $x()$. For instance, we will write $x + 3$ for $x() + 3$. We assume a function $ID : \mathcal{T} \to 2^F$ that, applied to a term $\tau \in \mathcal{T}$, returns the set of the function symbols appearing in τ.

Example 1. Let $ar(a) = ar(b) = 1$ and x, y be two real variables. If τ_1 is the term $\frac{4}{9} \cdot x \cdot (b(x + a(y^2)))^4$ and τ_2 is $9 + 3 \cdot x + \frac{2}{3} \cdot y + b(a(y))$ then τ_1 is a polynomial term, τ_2 is a linear term, and $ID(\tau_1) = ID(\tau_2) = \{x, y, a, b\}$.

For a term $\tau \in \mathcal{T}$ and a valuation $v \in \mathcal{V}$, we denote with $[\tau]_v$ the value of τ under the valuation v, more precisely:

$$\begin{aligned} [q]_v &= q \\ [q \cdot a(\tau_1, \ldots, \tau_{ar(a)})]_v &= q \cdot v(a)([\tau_1]_v, \ldots, [\tau_{ar(a)}]_v) \\ [\tau_1 + \tau_2]_v &= [\tau_1]_v + [\tau_2]_v \\ [\tau_1 \cdot \tau_2]_v &= [\tau_1]_v \cdot [\tau_2]_v \end{aligned}$$

Definition 2. *The set Φ of* quantified formulae *over $\mathcal{T}$ is defined inductively by:*

$$\phi ::= \tau \sim \tau' \mid int(\tau) \mid \exists a \,.\, \phi' \mid \neg\phi_1 \mid \phi_1 \vee \phi_2$$

where $\tau, \tau' \in \mathcal{T}$, $\sim \in \{<, \leq, =, \geq, >\}$ and $a \in F$. Conjunction, implication and universal quantification can be easily derived. We say that ϕ is linear *iff for each $\tau \sim \tau'$ and $int(\tau)$ appearing in ϕ, it holds that τ, τ' are linear terms.*

For a quantified formula $\phi \in \Phi$ and a valuation $v \in \mathcal{V}$, we write $v \models \phi$ if v *satisfies* ϕ. The relation $\models$ is defined as follows:

$$\begin{aligned} v \models \tau \sim \tau' &\text{ iff } [\tau]_v \sim [\tau']_v \\ v \models int(\tau) &\text{ iff } [\tau]_v \in \mathbb{Z} \\ v \models \exists a.\phi' &\text{ iff there exists a } f : \mathbb{R}^{ar(a)} \to \mathbb{R} \text{ s.t. } v[f/a] \models \phi' \\ v \models \neg\phi_1 &\text{ iff } v \not\models \phi_1 \text{ (namely } v \models \phi \text{ does not hold)} \\ v \models \phi_1 \vee \phi_2 &\text{ iff } v \models \phi_1 \text{ or } v \models \phi_2. \end{aligned}$$

For a formula $\phi \in \Phi$, we denote with $[\![\phi]\!]$ the set of valuations in $\mathcal{V}$ that satisfy ϕ, formally $[\![\phi]\!] = \{v \in \mathcal{V} \mid v \models \phi\}$. Two formulae ϕ_1 and ϕ_2 are *equivalent* iff $[\![\phi_1]\!] = [\![\phi_2]\!]$. We denote with *true* the formula $0 = 0$ and with *false* the formula $\neg true$. So, $true, false$ are in Φ. Notice that $[\![false]\!] = \emptyset$ and $[\![true]\!] = \mathcal{V}$. Moreover, with $Free(\phi)$ we denote the set of real variables and functions symbols non-quantified in ϕ, and with $Bnd(\phi)$ we denote the set of real variables and functions symbols quantified in ϕ. We assume, without loss of generality, that the sets $Free(\phi)$ and $Bnd(\phi)$ are always disjoint.

Example 2. Let x, y be two real variables, and a, b be two function symbols with $ar(a) = 1$ and $ar(b) = 2$. We give some properties that can be expressed in Φ:

- The function $b(\_, 1)$ is the square of $a(\_)$: $\forall x.a(x) \cdot a(x) = b(x, 1)$.
- The function a assumes only integer values: $\forall x.int(a(x))$.
- The function a reaches x: $\exists y.a(y) = x$.
- The minimum of a is in position x: $\forall y.a(y) \geq a(x)$.
- The function a is binary: $\forall x.a(x) = 0 \vee a(x) = 1$.
- The function a is monotone: $\forall x, y.x \leq y \Rightarrow a(x) \leq a(y)$.

Given two terms $\tau_1, \tau_2 \in \mathcal{T}$, with $\phi[\tau_1 := \tau_2]$ we denote the formula obtained by replacing all occurrences of τ_1 in ϕ with τ_2. Moreover, if a is a function symbol in F, let $App(a, \phi)$ denote the set of terms $a(\tau_1, \ldots, \tau_{ar(a)})$ appearing in ϕ.

Definition 3. $\mathcal{E} = \{\exists a. \mid a \in F\}^*$ *is called the set of* existential quantification prefixes*,* $\mathcal{F} = \{\forall a. \mid a \in F\}^*$ *is called the set of* universal quantification prefixes*, and* $\mathcal{Q} = \{\exists a., \forall a. \mid a \in F\}^*$ *is called the set of* quantification prefixes*.*

For instance, $\exists a.\exists b.$ is in $\mathcal{E}$ and $\exists a.\forall b.$ is in $\mathcal{Q}$. From now on, without loss of generality we consider formulae of the form $Q.\phi$ where $Q \in \mathcal{Q}$, and $Bnd(\phi) = \emptyset$.

Definition 4. *A quantified formula* $\phi \in \Phi$ *is* satisfiable *iff* $[\![\phi]\!] \neq \emptyset$*. The* satisfiability problem *for* $\phi \in \Phi$ *consists in answering if* ϕ *is satisfiable.*

The satisfiability problem is in general undecidable if one considers quantified polynomial formulae on integer variables (see [23]). Different classes of formulae have been considered to avoid such a difficulty. For quantified linear formulae on real and integer variables (without function symbols a with $ar(a) > 0$), the satisfiability problem is decidable (see [34]). In fact, in this case quantifiers can be eliminated through *quantifier elimination*, which consists in transforming a formula $\exists x.\phi$ where $Bnd(\phi) = \emptyset$ into an equivalent formula ϕ' s.t. $Bnd(\phi') = \emptyset$. For instance, $\exists x.5 < x \wedge x < y$ can be transformed into the equivalent formula $5 < y$. In general, quantifier elimination does not hold if one considers also functions (see [7] and [16] for example). The problem is that the variables appearing in the scope of some functions cannot be eliminated. The following proposition states that quantifier elimination cannot work, in general, for the formulae in Φ.

Proposition 1. *Let* ϕ *be the formula* $\forall x\,.\,0 \leq x < y \Rightarrow a(x) = 0$*. There is no formula* $\phi' \in \Phi$ *with* $Bnd(\phi') = \emptyset$ *equivalent to* ϕ*.*

For a formula $\exists a.\phi$ with $Bnd(\phi) = \emptyset$, we say that a is an *uninterpretated function*. The satisfiability of such a formula is decidable. In fact, by following [30], one can get a formula ϕ' equivalent to $\exists a.\phi$ by substituting each occurrence $a(\tau_1, \ldots, \tau_{ar(a)})$ in ϕ with a real variable x, provided ϕ is updated in a suitable way that, however, does not require to introduce any function with arity > 0. Then, since the ϕ' so obtained does not contain functions with arity > 0, its satisfiability is decidable according to [34]. For instance, $\exists a.a(4) \leq a(k)$ can be turned into $\exists x.\exists y.x \leq y \wedge ((4 = k) \Rightarrow (x = y))$. The challenge is to extend this method to formulae not in the form $\exists a.\phi$ with $Bnd(\phi) = \emptyset$. This is immediate for formulae of the form $\exists a.\exists x.\phi$ with $Bnd(\phi) = \emptyset$, since such a formula is equivalent to $\exists x.\exists a.\phi$, which can be transformed to $\exists x.\phi'$, to which quantifier elimination applies. The difficulty is in the formulae of the form $\exists a.\forall x.\phi$ and $\forall a.\exists x.\phi$. In next section we consider the case $\exists a.\forall x.\phi$ since $\forall a.\exists x.\phi = \neg\exists a.\forall x.\neg\phi$.

3 The function elimination method

In this section we give two results showing under which circumstances a function in a quantified formula in Φ can be replaced by a function with less arguments. If, by iteratively applying these substitutions, we transform each function into a real variable, then satisfiability follows from results in [32] and [34].

First of all we investigate when all occurrences in a formula ϕ of a function a can be substituted by occurrences of a function b that applies to a subset of the arguments of a (Theorem 1 below).

Definition 5. *A* term $\tau \in \mathcal{T}$ is determined by a real variable x *iff* $[\tau]_v \neq [\tau]_{v'}$ *for all valuations* $v, v' \in \mathcal{V}$ *s.t.* $v(x) \neq v'(x)$, *and* $v(y) = v'(y)$ *for all* $y \neq x$.

For instance, the terms $x^3 + a(z) \cdot z^2$ and $3 \cdot x + 2$ are determined by x, whereas the terms x^2 and $a(x)$ are not determined by x.

Proposition 2. *It is decidable to check whether* τ *is determined by* x.

Definition 6. *Let* $a \in F$, $S \subseteq \{1, \ldots, ar(a)\}$ *with* $\mathcal{S} \neq \emptyset$, x *be a real variable and* $\phi \in \Phi$. *We say that* a is $\mathcal{S}$-fixed on x in ϕ *iff:*

- *for each* $i \notin \mathcal{S}$ *and* $a(\tau_1, \ldots, \tau_{ar(a)}) \in App(a, \phi)$, *it holds that* $x \notin ID(\tau_i)$;
- *for each* $i \in \mathcal{S}$, *there exists a term* $\hat{\tau}_i$ *determined by* x *s.t.* $ID(\hat{\tau}_i) \subseteq Free(\phi) \cup \{x\}$ *and, for all* $a(\tau_1, \ldots, \tau_{ar(a)}) \in App(a, \phi)$, *it holds that* $\tau_i = \hat{\tau}_i$.

Informally, a is $\mathcal{S}$-fixed on x in ϕ iff for all terms of the form $a(\tau_1, \ldots, \tau_{ar(a)})$ appearing in ϕ, the real variable x appears in all and only the terms τ_i with $i \in \mathcal{S}$ and, for each $i \in \mathcal{S}$, the respective term τ_i is unique and determined by x. This implies that, for each $c, c' \in \mathbb{R}$ with $c \neq c'$, the values of the argument terms τ_i with $i \in \mathcal{S}$ to which a is applied in $\phi[x := c]$ are different from those to which a is applied in $\phi[x := c']$. Hence, $\phi[x := c]$ and $\phi[x := c']$ can be considered separately if one wants to delete the quantified function symbol a.

Example 3. We consider the following running example. Let ϕ_r be the formula $\exists b.\forall x.\forall y.\exists a.\forall z.a(z + b(y^3, x + 5)) = b(y^3, b(y^3, 3)) \wedge a(0) = k$ where x, y, z and k are real variables, and a, b are functions s.t. $ar(a) = 1$ and $ar(b) = 2$. We have that b is $\{1\}$-fixed on y in ϕ_r. Then, b is not $\mathcal{S}$-fixed on x in ϕ_r, for any $\mathcal{S}$. Finally, a is not $\{1\}$-fixed on any variable in ϕ_r.

If a is $\mathcal{S}$-fixed on x in ϕ with $\{1, \ldots, ar(a)\} \setminus \mathcal{S} = \{i_1, \ldots, i_m\}$, and a' is a function symbol s.t. $ar(a') = m$, then let $Sub(a, a', \mathcal{S}, \phi)$ denote the formula

$$(\phi)\{[a(\tau_1, \ldots, \tau_{ar(a)}) := a'(\tau_{i_1}, \ldots, \tau_{i_m})] \mid a(\tau_1, \ldots, \tau_{ar(a)}) \in App(a, \phi)\}$$

resulting from the substitution in ϕ of each occurrence of a with a' applied to the argument terms of a that are not in positions in $\mathcal{S}$.

Theorem 1. *Assume a formula $\phi \in \Phi$, a function symbol $a \in F$, a real variable x and a set $\mathcal{S} \subseteq \{1, \ldots, ar(a)\}$. If a is $\mathcal{S}$-fixed on x in ϕ, then*

$$\exists a.\forall x.\phi \;\; \textit{is equivalent to} \;\; \forall x.\exists a'.Sub(a, a', \mathcal{S}, \phi).$$

Example 4. Consider the formula ϕ_r in Ex. 3. By applying Th. 1 to b we get the equivalent formula ϕ'_r: $\forall y.\exists b'.\forall x.\exists a.\forall z.a(z + b'(x + 5)) = b'(b'(3)) \wedge a(0) = k$.

Now we investigate when a given term $a(\tau_1, \ldots, \tau_{ar(a)})$ in $App(a, \phi)$ can be replaced by a term $a'(\tau_1, \ldots, \tau_{j-1}, \tau_{j+1}, \ldots \tau_{ar(a)})$. The idea is to generalize to arbitrary functions the technique proposed in [30] to replace uninterpreted functions with functions with less arguments.

Let $\phi \in \Phi$ and $a \in F$. With $EQ(a, \phi)$ we denote the formula

$$\bigwedge_{\substack{a(\tau_1,\ldots,\tau_{ar(a)}) \\ a(\tau'_1,\ldots,\tau'_{ar(a)})}} \in App(a,\phi) \; \bigwedge_{i \in [1, ar(a)]} \tau_i = \tau'_i \Rightarrow a(\tau_1, \ldots, \tau_{ar(a)}) = a(\tau'_1, \ldots, \tau'_{ar(a)})$$

expressing that if two arbitrary occurrences of a in ϕ are applied to argument terms with the same values, then the resulting values are equal.

Example 5. Let ϕ'_r be as the formula in Ex. 4. Then $EQ(b', \phi'_r)$ is equivalent to $(x+5 = b'(3)) \Rightarrow b'(x+5) = b'(b'(3)) \wedge (x+5 = 3) \Rightarrow b'(x+5) = b'(3) \wedge (b'(3) = 3) \Rightarrow b'(b'(3)) = b'(3)$.

In [2] it is proved that the formula $\exists f.\forall y.P(y, f(y))$, with P a predicate, is equivalent to $\forall y.\exists x.P(y, x)$. We apply the same strategy to formulae in Φ.

Theorem 2. *Let $\phi' \in \Phi$ be the formula $\exists a.Q.\phi$ with $Bnd(\phi) = \emptyset$, and let $a(\tau_1, \ldots, \tau_{ar(a)})$ be a term in $App(a, \phi)$ s.t. for some $1 \leq j \leq ar(a)$ it holds $ID(\tau_j) \subseteq Free(\phi')$. Then ϕ' is equivalent to the formula*

$$\exists a.\exists a'.Q.(\phi \wedge EQ(a, \phi))[a(\tau_1, \ldots, \tau_{ar(a)}) := a'(\tau_1, \ldots, \tau_{j-1}, \tau_{j+1}, \ldots, \tau_{ar(a)})].$$

Example 6. Let us consider the formula ϕ'_r in Ex. 4. By applying Th. 2 to term $b'(3) \in App(b', \phi'_r)$ we obtain the equivalent formula
$\forall y.\exists b'.\exists w.\forall x.\exists a.\forall z.a(z + b'(x + 5)) = b'(w) \ \wedge \ a(0) = k \ \wedge \ (w = 3 \Rightarrow b'(w) = w) \ \wedge \ (x + 5 = w \Rightarrow b'(x + 5) = b'(w)) \ \wedge \ (x + 5 = 3 \Rightarrow b'(x + 5) = w)$.
Now, by applying Th. 2 to term $b'(w)$, we get the equivalent formula
$\forall y.\exists b'.\exists w.\exists t.\forall x.\exists a.\forall z.a(z + b'(x + 5)) = t \ \wedge \ a(0) = k \ \wedge \ (w = 3 \Rightarrow t = w) \ \wedge \ (x + 5 = w \Rightarrow b'(x + 5) = t) \ \wedge \ (x + 5 = 3 \Rightarrow b'(x + 5) = w)$.
We can now apply Th. 1 to function symbol b', which is now $\{1\}$-fixed on x, thus obtaining the equivalent formula
$\forall y.\exists w.\exists t.\forall x.\exists s.\exists a.\forall z.a(z + s) = t \ \wedge \ a(0) = k \ \wedge \ (w = 3 \Rightarrow t = w) \ \wedge \ (x + 5 = w \Rightarrow s = t) \ \wedge \ (x + 5 = 3 \Rightarrow s = w)$.
By applying Th. 2 to term $a(0)$ we get the equivalent formula
$\forall y.\exists w.\exists t.\forall x.\exists s.\exists a.\exists a'.\forall z.a(z + s) = t \ \wedge \ a' = k \ \wedge \ (w = 3 \Rightarrow t = w) \ \wedge \ (x + 5 = w \Rightarrow s = t) \ \wedge \ (x + 5 = 3 \Rightarrow s = w) \ \wedge \ (z + s = 0 \Rightarrow a(z + s) = a')$.
By applying Th. 1 to function symbol a, which is $\{1\}$-fixed on z, we obtain the equivalent formula
$\forall y.\exists w.\exists t.\forall x.\exists s.\exists a'.\forall z.\exists a''.a'' = t \ \wedge \ a' = k \ \wedge \ (w = 3 \Rightarrow t = w) \ \wedge \ (x + 5 = w \Rightarrow s = t) \ \wedge \ (x + 5 = 3 \Rightarrow s = w) \ \wedge \ (z + s = 0 \Rightarrow a'' = a')$.
which has no function with arity > 0. Therefore its satisfiability is decidable. The formula above is equivalent to the original formula ϕ_r in Ex. 3.

The following result is a generalization of Th. 2.

Corollary 1. *Let $\phi' \in \Phi$ be the formula $\exists a.Q.\phi$ with $Bnd(\phi) = \emptyset$, and let $a(\tau_1, \ldots, \tau_{ar(a)})$ be a term in $App(a, \phi)$ s.t. for some $j_1, \ldots, j_k$ in $1, \ldots, ar(a)$, $ID(\tau_{j_h}) \subseteq Free(\phi')$ for each $h = 1, \ldots, k$. Then ϕ' is equivalent to the formula*

$$\exists a.\exists a'.Q.(\phi \wedge EQ(a, \phi))[a(\tau_1, \ldots, \tau_{ar(a)}) := a'(\tau_{i_1}, \ldots, \tau_{i_{ar(a)-k}})]$$

where $ar(a') = ar(a) - k$ and $\{i_1, \ldots, i_{ar(a)-k}\} = \{1, \ldots, ar(a)\} \setminus \{j_1, \ldots, j_k\}$.

4 The sets Φ_L and Φ_P

In this section we characterize the subset of formulae in Φ for which we can prove that satisfiability is decidable by means of Theorems 1 and 2.

For a formula $\phi \in \Phi$, we say that *the variable x is in the scope of function a* iff there is a term $a(\tau_i, \ldots, \tau_{r(f)}) \in App(a, \phi)$ s.t. x appears in some τ_i.

We say that $\phi \in \Phi$ is a *basic formula* iff it has the form $Q.\phi'$, where $Q \in \mathcal{Q}$, $Bnd(\phi') = \emptyset$ and no function symbol a with $ar(a) > 1$ is universally quantified by Q. Hence, for some ϕ' with $Bnd(\phi') = \emptyset$, $E_1, \ldots, E_{n+1}$ in $\mathcal{E}$ (recall that $\varepsilon \in \mathcal{E}$), and real variables $x_1, \ldots, x_n$, a basic formula has the form

$$E_1.\forall x_1. \ldots E_n.\forall x_n.E_{n+1}.\phi'.$$

For a basic formula $\exists a.E_1.\forall x_1. \ldots E_n.\forall x_n.E_{n+1}.\phi$, the following definition gives a condition ensuring that, by iteratively applying Th. 1, all occurrences of $a(\tau_1, \ldots, \tau_{ar(a)}) \in App(a, \phi)$ can be substituted by terms $a'(\tau_{i_1}, \ldots, \tau_{i_m})$ s.t. no universally quantified variable x_i is in the scope of a'.

Definition 7. *Given the basic-formula* $\phi = \exists a.E_1.\forall x_1 \dots E_n.\forall x_n.E_{n+1}.\phi'$*, let* j *be the maximal index in* $\{1,\dots,n\}$ *s.t.* x_j *is in the scope of* a*. Then we say that* a *is a* sequenced-function*, denoted as S-function for short, iff for each index* $i \in \{1,\dots,j\}$ *there exists a set of indexes* $\mathcal{S}_{x_i} \neq \emptyset$ *s.t.* a *is* $\mathcal{S}_{x_i}$*-fixed on* x_i *in* ϕ*. Moreover, for each term* $a(\tau_1,\dots,\tau_{ar(a)}) \in App(a,\phi')$*, for all argument terms* τ_h *we have either:*

- $ID(\tau_h) \subseteq Free(\phi)$*,*
- *or* $x_l \in ID(\tau_h)$ *for some* $l \in \{1,\dots,j\}$*.*

The idea behind Def. 7 is that $\phi = \exists a.E_1.\forall x_1 \dots E_n.\forall x_n.E_{n+1}.\phi'$ can be mapped through j application of Th. 1 to an equivalent formula of the form $E_1.\forall x_1 \dots E_j.\forall x_j \exists a'.E_{j+1}.\forall x_{j+1} \dots E_n.\forall x_n.E_{n+1}.\phi''$, where all occurrences of $a(\tau_1,\dots,\tau_{ar(a)}) \in App(a,\phi')$ have been replaced by occurrences of $a'(\tau_{i_1},\dots,\tau_{i_m})$ in ϕ'' s.t. no universally quantified x_i is in the scope of a'.

Example 7. The formula $\exists b.\forall x.\exists a.\forall y.\ x < y \Rightarrow a < b(y)$ does not respect Def. 7, since y is in the scope of b and b is not $\{1\}$-fixed on x. Since b is not $\{1\}$-fixed on x, we cannot apply Th. 1 to b and x and the formula cannot be transformed by applying Th. 1.

The formula $\exists a.\exists c.\forall x.\forall y.\ x < y \Rightarrow a(x) < c(y,x)$ respects Def. 7, since a is $\{1\}$-fixed on x and c is $\{2\}$-fixed on x and $\{1\}$-fixed on y. By applying Th. 1 to a and x we get the equivalent formula $\exists c.\forall x.\exists a'.\forall y.\ x < y \Rightarrow a' < c(y,x)$. Then, by applying Th. 1 to c and x we get the equivalent formula $\forall x.\exists a'.\exists c'.\forall y.\ x < y \Rightarrow a' < c'(y)$. Finally, by applying Th. 1 to c' and y, we get the formula $\forall x.\exists a'.\forall y.\exists c''.\ x < y \Rightarrow a' < c''$ with only real variables.

Our aim is now to formally define a sequenced formula as a formula in Φ in which all functions are sequenced functions. For the basic formula ϕ of the form $E_1.\forall x_1 \dots E_n.\forall x_n.E_{n+1}.\phi'$, we denote with $Ex(i,\phi)$ the set of existentially quantified function symbols that occur in E_i. Formally, we define $Ex(i,\phi) = \{a_1,\dots,a_m\}$ if E_i has the form $\exists a_1.\exists a_2 \dots \exists a_m$; in particular we define $Ex(1,\phi) = \{a_i \mid a_i \text{ is existentially quantified in } E_1\} \cup Free(\phi)$.

Definition 8. *The set of* sequenced-formulae*, S-formulae for short, is the least set closed under disjunction and conjunction that contains the basic formula* $E_1.\forall x_1 \dots E_n.\forall x_n.E_{n+1}.\phi$ *with* $a \in Ex(i,\phi)$ *for some* $i = 1,\dots,n+1$ *iff* a *is an S-function for the subformula* $\exists a.E_i.\forall x_i \dots E_n.\forall x_n.E_{n+1}.\phi$*.*

With Φ_L we denote the set of S-formulae that are linear. With Φ_P we denote the set of S-formulae ϕ s.t. no occurrence of predicate $int(\_)$ appears. For instance, all formulae in Ex. 2 but the first, are in Φ_L. Moreover, the formula in Ex. 3 and the first formula of Ex. 2 are in Φ_P.

Proposition 3. *For each S-formula* ϕ*, there exists a S-formula* ϕ' *disjunction of basic formulae that is equivalent to* ϕ*.*

From now on we suppose that each formula is of the form as in Proposition 3. As a result of Def. 7 and Def. 8, if ϕ is an S-formula, then each value assumed

by a function symbol a, that as to be an S-function, can be related with itself and a fixed finite set of values assumed by a, whichever is its arity. The following theorem proves that satisfiability is decidable for formulae in Φ_L and Φ_P.

Theorem 3. *The satisfiability problem is decidable for Φ_L and Φ_P.*

We now aim to show that the sets Φ_L and Φ_P of S-formulae are the biggest decidable classes. In other terms, all requirements of Def. 7, and so of Def. 8, are necessary to ensure that satisfiability is decidable. The result of [7] can be used to prove that, if we relax the definition of S-formulae, the satisfiability problem for Φ_L and Φ_P is undecidable. In fact, [7] uses functions to calculate the satisfiability of formula with polynomial terms and integer variables.

Firstly we focus on Φ_L. For instance, we can consider the generic polynomial term $\tau \cdot h$, where τ is a linear term and h is an integer variable. Let $\phi_{(sol,\tau,h)}$ be the linear formula $int(sol) \wedge sol = a(\tau+h) \wedge a(\tau+1) = \tau \wedge \forall\, l \in [\tau+1, \tau+h]\,.\, a(l+1) = a(l) + \tau$. Notice that this formula is not an S-formula, since function symbol a is not S_l-fixed on l, $\forall\, l \in [\tau+1, \tau+h]$.

The value $\tau \cdot h$ is equal to the integer value assumed by the variable sol which satisfies $\phi_{(sol,\tau,h)}$. In fact, $sol = a(\tau+h) = a(\tau+h-1)+\tau = a(\tau+h-2)+2\cdot\tau = \cdots = a(\tau+1) + (h-1)\cdot\tau = \tau + (h-1)\cdot\tau = h \cdot \tau$.

By applying this technique recursively, a value of a polynomial term τ can be easily written as an integer value assumed by a variable which satisfies a linear formula. For instance, $\tau \cdot h_1 \cdot h_2$ can be written as $sol_1 \cdot h_2$, where sol_1 satisfies the formula $\phi_{(sol_1,\tau,h_1)}$. Now sol_1 is trivially a linear term, and hence, $\tau \cdot h_1 \cdot h_2$ is equal to the value of sol_2 of the valuations which satisfy $\phi_{(sol_1,\tau,h_1)} \wedge \phi_{(sol_2,sol_1,h_2)}$. As a consequence, a polynomial formula $\tau \sim 0$ can be written as an equivalent linear formula. Hence the satisfiability problem for Φ_L is undecidable if we relax the definition of S-functions and, consequently, of S-formulae, by removing the S-fixed property.

We now take on the case of Φ_P. The satisfiability of a formula with polynomial terms and real variables is decidable. But we can use the functions to force a real variable to be an integer. For instance, let $\tau \sim 0$ be a formula on integer variables; we can write a formula ϕ, with no occurrences of predicate $int(\_)$, s.t. $\tau \sim 0$ is satisfiable iff ϕ is.

Let $\{k_1, \ldots, k_n\}$ be the integer variables which appear in τ, and $\{x_1, \ldots, x_n\}$ be a set of real variables. Let $\tau' = \tau[k_1 := x_1] \ldots [k_n := x_n]$. It is sufficient to consider as the formula ϕ the following formula:
$\tau' \sim 0 \wedge \bigwedge_{i\in[1,n]} (x_i \geq 0 \Rightarrow a(x_i) = x_i) \wedge (x_i < 0 \Rightarrow a(-x_i) = -x_i) \wedge \forall y.(0 \leq y \wedge y < 1 \Rightarrow a(y) = 0) \wedge (y \geq 1 \Rightarrow a(y) = a(y-1) + 1)$.

We notice that, as in the previous case, function symbol a, which expresses the integer part of a real variable, is not an S-function, since it is not S_y-fixed on y in ϕ. Thus, $\phi \notin \Phi_P$. We are going to show that, for each $v \models \phi$, if x_i is positive, then $a(x_i) = x_i$ iff x_i is a natural. If this holds, then the satisfiability problem for ϕ is undecidable and therefore we cannot relax the definition of S-formula without loosing decidability.

Let us suppose that $v \models \phi$ and $a(x_i) = x_i$, then there exist $\epsilon \in [0, 1)$ and $k \in \mathbb{N}$ s.t. $v(a(x)) = v(a(\epsilon)) + k$. But, for each $y < 1$, it holds that $a(y) = 0$,

and therefore $v(a(\epsilon)) = 0$. Hence $v(x) = v(a(x)) = v(a(\epsilon)) + k = k$, for some natural k. Conversely, if v is a valuation where $v \models \phi$ and $v(x)$ is a natural, then $a(v(x)) = a(v(x)-1)+1$. Since $v(x)$ is a natural value, $v(a(x)) = v(a(0))+v(x) = 0 + v(x) = v(x)$. Hence $a(x) = x$. Now it is obvious that if x_i is negative, then it is an integer iff $a(-x_i) = -x_i$. So we can conclude as above that satisfiability is not decidable for formulae not in Φ_P.

Examples studied above, show that requirements of Def. 7, and so of Def. 8, are necessary to ensure that satisfiability is decidable. Thus, the sets Φ_L and Φ_P are the biggest decidable classes.

5 Modeling sets with Φ_L and Φ_P

The sets of formulae Φ_L and Φ_P are sufficient to model sets and strings.

In the following we show how to write classical operators and properties on sets of integers and reals, where A and B are sets of n-tuples.

- $A = \{(x_1, \ldots, x_n) \,|\, \phi\}$ where ϕ is a formula with only real variable, can be expressed with the formula $\forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) = 1 \Leftrightarrow \phi$
- $(\tau_1, \ldots, \tau_n) \in A$ can be expressed by the formula $a_A(\tau_1, \ldots, \tau_n) = 1$
- $A = B \cap C$ can be expressed by the formula $\forall x_1, \ldots, x_n .\ a_A(x_1, \ldots, x_n) = 1 \Leftrightarrow a_B(x_1, \ldots, x_n) = 1 \wedge a_C(x_1, \ldots, x_n) = 1$
- $A = B \cup C$ can be expressed by the formula $\forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) = 1 \Leftrightarrow a_B(x_1, \ldots, x_n) = 1 \vee b_C(x_1, \ldots, x_n) = 1$
- $A = B \setminus C$ can be expressed be the formula $\forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) = 1 \Leftrightarrow a_B(x_1, \ldots, x_n) = 1 \wedge a_C(x_1, \ldots, x_n) \neq 1$
- $A \subseteq B$ can be expressed by the formula $\forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) = 1 \Rightarrow a_B(x_1, \ldots, x_n) = 1$
- $|A| = 0$ can be expressed by the formula $\forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) \neq 1$
- $|A| \leq k$ can be expressed by the formula $\exists y_1^1 . \exists y_2^1 \ldots \exists y_n^k . \forall x_1 \ldots \forall x_n . a_A(x_1, \ldots, x_n) = 1 \Rightarrow \bigvee_{i \in [1,k]} \bigwedge_{j \in [1,n]} x_j = y_j^i$
- $|A| \geq k$ can be expressed by the formula $\exists x_1^1 . \exists x_2^1 \ldots \exists x_n^k . \bigwedge_{i \in [1,k]} a_A(x_1^i, \ldots, x_n^i) = 1 \wedge \bigwedge_{j \neq i} \ \bigwedge_{h \in [1,n]} x_h^i \neq x_h^j$.

We note that the formulae above are closed on conjunction and disjunction.

Example 8. The formula $\exists B . \exists C \exists z . A = \{x \,|\, \exists w . w^2 x^3 + 2x + 1 \leq z\} \wedge B = A \setminus C \wedge |B| \neq 0$ is satisfiable iff $\exists a_B . \exists . a_C \exists . z . \exists y . \forall x . \exists w . (a_A(x) = 1 \Leftrightarrow w^2 x^3 + 2x + 1 \leq z) \wedge (a_B(x) = 1 \Leftrightarrow a_A(x) = 1 \wedge a_C(x) \neq 1) \wedge a_B(y) = 1$ This last formula is in Φ_P, and therefore the satisfiability is decidable.

In the example above we have considered sets of reals where formulae to express properties are polynomial. If one wants to consider also integers one must restrict to linear formulae.

Example 9. The formula $\exists B . \exists C . \exists z . A = \{(x, y) \,|\, int(x + y) \wedge z = 2x\} \wedge B = A \setminus C \wedge B \neq \emptyset$ is satisfiable iff $\exists a_B . \exists a_C . \exists z . \forall x . \forall y . \exists w_1 . \exists w_2 . a_A(x, y) = 1 \Leftrightarrow int(x + y) \wedge z = 2x) \wedge ((a_B(x, y) = 1 \Leftrightarrow a_A(x, y) = 1 \wedge a_C(x, y) \neq 1) \wedge a_B(w_1, w_2) = 1$ This last formula is in Φ_L, and therefore the satisfiability is decidable.

6 Conclusions and Related works

We have proved that, in the framework of Arithmetical Logics, quantifier elimination for formulae with functions cannot exist, and hence we have defined a technique for decreasing the arity of functions appearing in a given formulae. We have proved the correctness of the method and we have used it to prove the decidability of the satisfiability problem for two classes of formulae. These classes have sufficient power to model theories as those of sets and strings.

In the literature several classes and methods have been studied. In [4] the decidable classes are surveyed. First of all, the paper considers classes that are subclasses of first order logics (hence no predicates and functions can be quantified). In [26] formulae with prefixes in $\exists^*\forall^*$ and without functions are considered. Formulae with prefixes in $\exists^*\forall^2\exists^*$ and without functions and equality predicates are tackled in [11], [24] and [27]. Formulae without equality predicate, and with unary predicate and functions are studied in [19] and [12]. In [22] and [13] prefixes in $\exists^*\forall\exists^*$ for formulae without equality predicates are considered, and in [14] existential quantified formulae are tackled. Finally universal quantified formulae with unary predicates and at most one unary function are studied in [3]. For all these classes, the finite model property is decidable, i.e. it is decidable to check whether there exists a model with a given finite complexity satisfying a given formula. The classes we define have a decidable result for generic model, moreover also quantifications on functions are considered. We do not consider predicates but these can be simulated by a function that can assume values in the set $\{0, 1\}$, where 0 represents $false$, and 1 represents $true$.

Monadic second order formulae are studied in [25] (see [15] and [33] for surveys). More precisely, a second order logic is defined (with unary predicates and at most one unary function) for which the satisfiability problem is decidable. There are several differences between this logic and those we consider. First of all, we have general arity but restrictions on the quantifications of variables and functions. Moreover, we consider reals and integers. In [25] only the natural field is considered (in [28] it is proved that the satisfiability problem for monadic second order logic on real field is undecidable). Finally we consider polynomial and general linear formulae, instead of formulae of the form $x < y$ as done in [25]. The class defined in [29] contains the set of second order formulae with prefixes in $\exists^*\forall\exists^*$, with predicates and one function. The classes we consider have a similar prefix, but the other differences shown for the class in [25] still hold.

Hence we must compare our classes with the classes allowing real and integer variables. Decidability results in this framework in usually given by a quantifier elimination technique. In [9] quantifier elimination for formulae which are linear and which contain real variables is studied. In [32] Tarski considers the general case of polynomial formulae on real variables. In the case of integer variables, if one considers linear formulae without quantifier on integer variables, then the satisfiability is decidable (see [36] and [18]). If one deals with First Order Logic on Linear Formulae with Integer Variables (called *Presburger Arithmetic*), [8] offers a quantifier elimination algorithm which is given by introducing equivalences modulo a natural value. In [10] it is shown that it is decidable in

double-exponential time. These first studies were directed to prove decidability. In recent years such quantifier elimination procedure has turned out to allow important applications e.g. in simulation and optimization, control theory and applied computational geometry ([1], [5], [6], [17] and [35]). Hence improvements to the efficiency of the algorithms are done in [20] and [31]. When one tackles formulae with mixed integer and real variables, techniques of quantifier elimination given for real field still hold. Conversely, these techniques do not hold for the case of integer variables. In [34], a quantifier algorithm for mixed variables is proposed. In this case, together with equivalences modulo a natural value, a function which gives the integer part of a real value is considered. All the classes mentioned consider first order logics without functions symbols. Hence they are subclasses of Φ_L and Φ_P. For formulae with exponential functions, which our logics cannot express, quantifier elimination exists for some cases (see [21]).

The satisfiability problem for High Order Arithmetical Logics (where function are quantifiable) is undecidable (due to the already mentioned undecidability result holding for polynomial formulae on integer variables). This bad result holds also if one considers linear formulae (see [7] and [16]). If uninterpreted functions are considered with linear formulae then the decidability holds [30]. It is immediate to see that the classes defined in this paper extend the class of uninterpreted functions by simply considering formulae of the form $\exists a.\forall x.\phi$.

References

1. Abdallah, C. T., Dorato, P., Yang, W., Liska, R., and Steinberg, S.: *Applications of Quantifier Elimination Theory to Control System Design.* In Proc. of the 4^{th} IEEE Mediterranean Symposium on Control and Automation (1996), IEEE, 340 – 345.
2. Ackermann, W.: *Solvable Cases of the Decision Problem.* North-Holland, Amsterdam, 1954.
3. Ash, C.: *Sentences with Finite Models.* Weitschr. f. math. Logik u. Grundlagen d. Math. **21** (1975), 401– 404.
4. Börger, E., Grädel, E., and Gurevich, Y.: *The Classical Decision Problem.* Springer, 1997.
5. Dolzmann, A., and Sturm, T.: *Redlog: Computer Algebra Meets Computer Logic.* ACM SIGSAM Bulletin 31 (1997), 2–9.
6. Dolzmann, A., Sturm, T. and Weispfenning, V.: *A New Approach for Automatic Theorem Proving in Real Geometry.* Journal of Automated Reasoning **21** (1998), 357-380.
7. Downey, P.: *Undecidability of Presburger Arithmetic with a Single Monadic Predicate Letter.* Technical Report 18-72, Center for Research in Computing Technology, Harvard Univ., 1972.
8. Enderton, H. B. .: *Mathematical Introduction to Logic.* Academic Press, 1972.
9. Ferrante, J., and Rackoff, C.: *A Decision Procedure for First-order Theory of Real Addition with Order.* SIAM Journal on Computing **4** (1975), 69–76.
10. Fischer, M. J., and Rabin, M. O.: *Super-Exponential Complexity of Presburger Arithmetic.* Complexity of Computation, SIAM-AMS Proc. **7** (1974), 27–42.
11. Godel, K.: *Ein Spezialfall des Entscheidungsproblems der theoretischen Logik.* Ergebn. math. Kolloq. **2** (1932), 27–28.

12. Gurevich, Y.: *The Decision Problem for the Logic of Predicates and Operations.* Algebra and Logic **8** (1969), 160–174.
13. Gurevich, Y.: *Formulas with one* $\forall$. Selected Questions in Algebra and Logics. Nauka, Novosibirsk, 1973, 97–110.
14. Gurevich, Y.: *The Decision Problem for Standard Classes.* Journal of Symbolic Computation **41** (1976), 460–464.
15. Gurevich, Y.: *Monadic Second-order Theories.* Model-theoretic Logics, Springer, Berlin, 1985, 479–506.
16. Halpern J. Y.: *Presburger Arithmetic with Unary Predicates is* Π_1^1 *Complete.* Journal of Symbolic Logic **56**, 1991, 637–642.
17. Honga, H., and Liskab, R. (Eds.): *Special Issue on Applications of Quantifier Elimination.* Journal of Symbolic Computation **24** (1997).
18. Lenstra H.W.: *Integer Programming with a Fixed Number of Variables.* Mathematics of Operations Research **8** (1983), 538–548.
19. Löb M.: *Decidability of The Monadic Predicate Calculus with Unary Function Symbols.* Journal of Symbolic Logic **32** (1967), pp. 563.
20. Loos, R., and Weipfenning, V.: *Applying Linear Quantifier Elimination.* The Computer Journal **36** (1993), 450–462.
21. Marker, D.: *Model Theory and Exponentiation.* Notices AMS **43** (1996), 753–759.
22. Maslov, S., and Orevkov, V.: *Decidable Classes Reducing to One-quantifier Class.* Proc. Steklov Ins. Steklov Math **121** (1972), 61–72.
23. Matijasevic, J. Y.: *Hilbert's Tenth Problem.* MIT press, Cambridge, 1993.
24. Kalmar, L.: *Uber die Erfullbarkeit derjenigen Zahlausdrucke, welehe in der Normalform zwei benachbarte Allzeichen enthalten.* Math. Ann. **108** (1933), 466-484.
25. Rabin, M.: *Decidability of Second Order Theories and Automata on Infinite trees.* Trans. Amer. Math. Soc. **141** (1969), 1–35.
26. Ramsey, F. P.: *On a Problem in Formal Logic.* Proc. of the London Mathematical Society **30** (1930), 264–286.
27. Schutte, K.: *Untersuchungen zum entscheidungsproblem der mathematischen logik.* Math. Annal. **109** (1934), 572–603.
28. Shelah, S.: *The Monadic Theory of Order.* Ann. of Math. **102** (1975), 379–419.
29. Shelah, S.: *Decidability of a Portion of the Predicate Calculus.* Israel Journal of Mathematics, **28** (1977), 32–44.
30. Shostak, R.E. : *A Practical Decision Procedure for Arithmetic with Functions Symbols.* Journal of ACM **26** (1979), 351–360.
31. Subrami, K.: *An Analysis of Quantified Linear Programs.* Proc. 4^{th} Int. Conference on Discrete Mathematics and Theoretical Computer Science, Lecture and notes in Computer Science, **2731** (2003), Springer, 265–277.
32. Tarski, A.: *A Decision Method for Elementary Algebra and Geometry.* University of California Press, Second Edition, 1951.
33. Thomas, W.: *Languages, Automata, and Logic.* Handbook of Formal Languages, **3** (1997), Springer Verlag, 389–455.
34. Weispfenning, V.: *Mixed Real-Integer Linear Quantifier Elimination.* Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation, 1999.
35. Weispfenning, V.: *A New Approach to Quantifier Elimination for Real Algebra.* In Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation, Springer, 1998, 376–392.
36. Williams, H. P.: *Fourier-Motzkin Elimination Extension to Integer Programming.* Journal of Combinatorial Theory **21** (1976), 118–123.