The Satisfiability of Word Equations: Decidable and Undecidable Theories

Joel D. Day[1(✉)], Vijay Ganesh[2], Paul He[2], Florin Manea[1], and Dirk Nowotka[1]

[1] Kiel University, Kiel, Germany
{jda,flm,dn}@informatik.uni-kiel.de
[2] University of Waterloo, Waterloo, Canada
vijay.ganesh@uwaterloo.ca, paul.he@edu.uwaterloo.ca

Abstract. The study of word equations is a central topic in mathematics and theoretical computer science. Recently, the question of whether a given word equation, augmented with various constraints/extensions, has a solution has gained critical importance in the context of string SMT solvers for security analysis. We consider the decidability of this question in several natural variants and thus shed light on the boundary between decidability and undecidability for many fragments of the first order theory of word equations and their extensions. In particular, we show that when extended with several natural predicates on words, the existential fragment becomes undecidable. On the other hand, the positive Σ_2 fragment is decidable, and in the case that at most one terminal symbol appears in the equations, remains so even when length constraints are added. Moreover, if negation is allowed, it is possible to model arbitrary equations with length constraints using only equations containing a single terminal symbol and length constraints. Finally, we show that deciding whether solutions exist for a restricted class of equations, augmented with many of the predicates leading to undecidability in the general case, is possible in non-deterministic polynomial time.

Keywords: Word equations · Decidability · Satisfiability

1 Introduction

A *word equation* is a formal equality $U = V$, where U and V are words (called the left and right side of the equation respectively) over an alphabet $A \cup X$; $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ is the alphabet of *constants* or *terminals* and $X = \{x_1, x_2, x_3, \ldots\}$ is the set of *variables*. A *solution* to the equation $U = V$ is a morphism $h : (A \cup X)^* \to A^*$ that acts as the identity on A and satisfies $h(U) = h(V)$; h is called the assignment to the variables of the equation. For instance, $U = x_1\mathtt{ab}x_2$ and $V = \mathtt{a}x_1x_2\mathtt{b}$ define the equation $x_1\mathtt{ab}x_2 = \mathtt{a}x_1x_2\mathtt{b}$, whose solutions are the morphisms h with $h(x_1) = \mathtt{a}^k$, for $k \geq 0$, and $h(x_2) = \mathtt{b}^\ell$, for $\ell \geq 0$. An equation is *satisfiable* (in A^*) if it admits a solution $h : (A \cup X)^* \to A^*$. A set (or system) of equations is satisfiable if there exists an assignment of the variables of the

I. Potapov and P.-A. Reynier (Eds.): RP 2018, LNCS 11123, pp. 15–29, 2018.
https://doi.org/10.1007/978-3-030-00250-3_2

equations in this set that is a solution for all equations. In logical terms, word equations are often investigated as fragments of the first order theory $\mathsf{FO}(A^*, \cdot)$ of strings. Karhumäki et al. [18] showed that deciding the satisfiability of a system of word equations, that is, checking the truth of formulas from the existential theory Σ_1 of $\mathsf{FO}(A^*, \cdot)$, can be reduced to deciding the satisfiability of a single (more complex) word equation that encodes the respective system.

The existential theory of word equations has been studied for decades in mathematics and theoretical computer science with a particular focus on the decidability of the satisfiability of logical formulae defined over word equations. Quine [28] proved in 1946 that the first-order theory of word equations is equivalent to the first-order theory of arithmetic, which is known to be undecidable. In order to solve Hilbert's tenth problem in the negative [14], Markov later showed a reduction from word equations to Diophantine equations (see [21,22] and the references therein), in the hopes that word equations would prove to be undecidable. However, Makanin [22] proved in 1977 that the satisfiability of word equations *is* in fact decidable. Though Markov's approach was unsuccessful, similar ones, based on extended theories of word equations, can also be explored. Matiyasevich [25] showed in 1968 a reduction from the more powerful theory of word equations with linear length constraints (i.e., linear relations between word lengths) to Diophantine equations. Whether this theory is decidable remains a major open problem. More than a decade after Makanin's decidability result, the focus shifted towards identifying the complexity of solving word equations. Plandowski [27] showed in 1999 that this problem is in PSPACE. Recently, in a series of papers (see specifically e.g., [15,16]), Jez applied a new technique called recompression to word equations. This lead to, ultimately, a proof that the satisfiability of word equations can be decided in linear space. However, there is a mismatch between this upper bound and the known lower bound: solving word equations is NP-hard, but whether the problem is NP-complete remains open.

In recent years, deciding the satisfiability of systems of word equations has also become an important problem in fields such as formal verification and security where string solvers such as HAMPI [19], CVC4 [3], Stranger [31], ABC [2], Norn [1], S3P [29] and Z3str3 [4] have become more popular. However, in practice more functionality than just word equations is required, so solvers often extend the theory of word equations with certain functions (e.g., linear arithmetic over the length, replace-all, extract, reverse, etc.) and predicates (e.g., numeric-string conversion predicate, regular-expression membership, etc.). Most of these extensions are not expressible by word equations, in the sense introduced by Karhumäki et al. [18], and some of them lead to undecidable theories. On the one hand, regular (or rational) constraints or constraints based on involutions (allowing to model the mirror image, or, when working with equations in free groups, inverse elements), are not expressible, see [6,18], but adding them to word equations preserves the decidability [8]. As mentioned above, whether the theory of word equations enhanced with a length function is decidable is still a major open problem. On the other hand, the satisfiability of word equations extended with a *replace-all* operator was shown to be undecidable in [20], and the same holds when a numeric-string conversion predicate is added. Due to this very complex and fuzzy picture, none of the solvers mentioned above has a complete algorithm.

Our Contributions: In this setting, our work aims to provide a better understanding of the boundary between extensions and restrictions of the theory of word equations for which satisfiability is decidable and, respectively, undecidable.

Firstly, we present a series of undecidability results for the Σ_1-fragment of $\mathsf{FO}(A^*, \cdot)$ extended with simple predicates or functions. In the main result on this topic, we show that extending Σ_1 with constraints imposing that a string is the morphic image of another one also leads to an undecidable theory. These results are related to the study of theories of quantifier-free word equations constrained by very simple relations, see, e.g., [6,13]. While our results do not settle the decidability of the theory of word equations with length constraints, they enforce the intuitive idea that enhancing the theory of word equations with predicates providing very little control on the combinatorial structure of the solutions of the equation leads to undecidability.

We further explore the border between decidability and undecidability when considering formulae over word equations allowing at most one quantifier alternation. We show that checking the truth of an arbitrary Σ_2-formula is equivalent to, on the one hand, checking the truth of a $\exists^*\forall^*$-quantified terminal-free formula, or, on the other hand, to a single $\exists^*\forall^*$-quantified inequation whose sides contain at most two terminals. Since the Inclusion of Pattern Languages problem (see [5,11,17]) can be reformulated as checking the truth of a single $\exists^*\forall^*$-quantified inequation whose sides contain at most two terminals and are variable disjoint, and it is undecidable, we obtain a clear image of the simplest undecidable classes of Σ_2-formulae. Consequently, we consider decidable cases. Complementary to the above, we show that the satisfiability in an arbitrary free monoid A^* of quantifier free *positive* formulae over word equations (formulae obtained by iteratively applying only conjunction and disjunction to word equations of the form $U = V$), in which we have at most one terminal $\mathsf{a} \in A$ (appearing zero or several times) and no restriction on the usage of variables, enhanced with linear length constraints, is decidable, and, moreover, NP-complete. The decidability is preserved when considering positive Σ_2-formulae of this kind, as opposed to the case of arbitrary Σ_2 terminal-free formulae, mentioned above. Moreover, if we allow negated equations in the quantifier-free formulae (so arbitrary Σ_1-formulae) with at most one terminal, and length constraints, we obtain a decidable theory if and only if the general theory of equations with length constraints is decidable. Putting together these results, we draw a rather precise border between the decidable and undecidable subclasses of the Σ_2-fragment over word equations, defined by restrictions on the number of terminals allowed to occur in the equations and the presence or absence of inequations. As a corollary, we can show that deciding the truth of arbitrary formulae from the positive Σ_2-fragment of $\mathsf{FO}(A^*, \cdot)$ (i.e., $\exists^*\forall^*$ quantified positive formulae), without length constraints, is decidable. The resulting proof follows arguments partly related to those in [9,23]. This result is strongly related to the work of [10,12,28], in which it was shown that the validity of sentences from the positive Π_2-fragment of $\mathsf{FO}(A^*, \cdot)$ (i.e., where the quantifier alternation was $\forall^*\exists^*$) is undecidable, as well as to the results of [30] in which it was shown that the truth of arbitrarily quantified positive formulae over word equations is decidable *over an infinite alphabet of terminals*.

We then extend our approach in a way partly motivated by the practical aspects of solving word equations. Most equations that can be successfully solved by string solvers (e.g., Z3str3) must be in *solved form* [12], or must not contain *overlapping variables* [32]. In a sense, this suggests that in practice it is interesting to find equations with restricted form that can be solved in reasonable time. We analyse, from a theoretical point of view, one of the simplest classes of equations that are not in solved form or contain equations with overlapping variables, namely strictly regular-ordered equations (each variable occurs exactly once in each side, and the order in which the variables occur is the same). We show that the satisfiability of such equations, even when enhanced with various predicates, is decidable. In particular we show that when extended with regular constraints (given by DFAs), linear length constraints, abelian equivalence constraints (two variables should be substituted for abelian-equivalent words), subword constraints (one variable should be a (scattered) subword of another), and $Eq_{\mathtt{a}}$ constraints (two variables should have the same number of occurrences of a letter $\mathtt{a}$), the satisfiability problem remains NP-complete. Thus, there is hope that they can be solved reasonably fast by string solvers based on, e.g., SAT-solvers. This line of results is also related to the investigations initiated in [7,24], in which the authors were interested in the complexity of solving equations of restricted form. In the most significant result of [7], it was shown that deciding the satisfiability of strictly regular-ordered equations (with or without regular constraints) is NP-complete, which makes this class of word equations one of simplest known classes of word equations that are hard to solve. Although these results regard a very restricted class of equations, they might provide some insights in tackling harder classes, such as, e.g., quadratic equations.

The organization of the paper is as follows. In Sect. 2 we introduce the basic notions we use. In Sect. 3, we present firstly the undecidability results related to theories over word equations extended with various simple predicates, secondly the undecidability and decidability results related to quantifier alternation, and thirdly, we present the results related to strictly regular-ordered equations. Due to space constraints, some proofs are omitted, or only briefly sketched.

2 Preliminaries

Let $\mathbb{N}$ be the set of natural numbers, and let $\mathbb{N}_{\leq n}$ be the set $\{1, 2, \ldots, n\}$. Let A be an alphabet of letters (or symbols). Let A^* be the set of all words over A and ε be the empty word. Note that A^* is a monoid w.r.t. the concatenation of words. Let $|w|$ denote the length of a word w and for each $a \in A$, let $|w|_a$ denote the number of occurrences of a in w. For $1 \leq i \leq |w|$ we denote by $w[i]$ the letter on the i^{th} position of w. A word w is *p-periodic* for $p \in \mathbb{N}$ (p is called a *period* of w) if $w[i] = w[i+p]$ for all $1 \le i \le |w| - p$; the smallest period of a word is called its *period*. If $w = v_1v_2v_3$ for some words $v_1, v_2, v_3 \in A^*$, then v_1 is called a *prefix* of w, v_1, v_2, v_3 are *factors* of w, and v_3 is a *suffix* of w. Two words w and u are called *conjugate* if there exist non-empty words v_1, v_2 such that $w = v_1v_2$ and $u = v_2v_1$. A word $v \in A^*$ is a *subword* of $w \in A^*$ if $v = v_1 \ldots v_k$, with $v_i \in A^*$, and $w = u_0v_1u_1 \cdots v_ku_k$, with $u_i \in A^*$. A word $z \in A^*$ is in the *shuffle* of $u, v \in A^*$, denoted $z \in u\Delta v$, if $z = u_1v_1 \cdots u_kv_k$, with $u_i, v_i \in A^*$, and $u = u_1 \cdots u_k$, $v = v_1 \cdots v_k$. Two words $u, v \in A^*$ are *abelian equivalent* if $|u|_a = |v|_a$, for all $a \in A$. The following lemma is well known (see, e.g., [21]).

Lemma 1 (Commutativity Equation). *Let $v_1, v_2 \in A^*$. Then $v_1v_2 = v_2v_1$ if and only if there exist $w \in A^*$ and $p, q \in \mathbb{N}_0$ such that $v_1 = w^p$ and $v_2 = w^q$.*

Let $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be a finite alphabet of *constants* and let $X = \{x_1, x_2, \ldots\}$ be an alphabet of *variables*. Note that we assume X and A are disjoint, and unless stated otherwise, that $|A| \ge 2$. A word $\alpha \in (A \cup X)^*$ is usually called a *pattern*. For a pattern α and a letter $z \in A \cup X$, let $|\alpha|_z$ denote the number of occurrences of z in α; $\mathrm{var}(\alpha)$ denotes the set of variables from X occurring in α. A morphism $h : (A \cup X)^* \to A^*$ with $h(a) = a$ for every $a \in A$ is called a *substitution*. A morphism $h : A^* \to B^*$ is a *projection* if $h(a) \in \{\epsilon, a\}$ for all $a \in A$. We say that $\alpha \in (A \cup X)^*$ is *regular* if, for every $x \in \mathrm{var}(\alpha)$, we have $|\alpha|_x = 1$; e. g., $\mathtt{a}x_1\mathtt{a}x_2\mathtt{c}x_3x_4\mathtt{b}$ is regular. Note that $L(\alpha) = \{h(\alpha) \mid h \text{ is a substitution}\}$ (the pattern language of α) is regular when α is regular.

A (positive) *word equation* is a tuple $(U, V) \in (A \cup X)^* \times (A \cup X)^*$; we usually denote such an equation by $U = V$, where U is the left hand side (LHS, for short) and V the right hand side (RHS) of the equation. A negative word equation, or inequation, is the negation of a word equation, i.e., $\neg(U = V)$ or $U \ne V$.

A *solution* to an equation $U = V$ (resp., $U \ne V$), over an alphabet A, is a substitution h mapping the variables of UV to words from A^* such that $h(U) = h(V)$ (respectively, $h(U) \ne h(V)$); $h(U)$ is called the *solution word*. Note that we might ask whether a positive or negative equation has a solution over an alphabet larger than the alphabet of terminals that actually occur in the respective equation. A word equation is *satisfiable* over A if it has a solution over A, and the *satisfiability problem* is to decide for a given word equation whether it is satisfiable over a given alphabet A.

Karhumäki et al. [18] have shown that, given two equations E and E', one can construct the equations E_1, E_2, and E_3 that are satisfiable in A^*, with $|A| \ge 2$, if and only if $E \wedge E'$, $E \vee E'$, $\neg E$ are satisfiable respectively in A^*. In this construction, E_1 contains exactly the variables of E and E', while in E_2 and E_3 new variables are added with respect to those in the given equations; in all cases, even if E and E' were terminal-free, the new equations contain terminals. We use this result to show that for every quantifier-free first order formula over word equations we can construct a single equation that may contain extra variables and terminals, and is satisfiable if and only if the initial formula was satisfiable. Moreover, the values the variables of the initial equations may take in the satisfying assignments of the new equation are exactly the same values they took in the satisfying assignments of the initial formula. We also use in several occasions the following result from [18].

Lemma 2. *Let* $U, V, U', V' \in (X \cup A)^*$, $Z_1 = U\mathtt{a}U'U\mathtt{b}U'$, $Z_2 = V\mathtt{a}V'V\mathtt{b}V'$. *For any substitution* $h : X^* \to A^*$, $h(Z_1) = h(Z_2)$ *iff* $h(U) = h(V)$ *and* $h(U') = h(V')$.

In this paper we address equations with restricted form. A word equation $U = V$ is regular if both U and V are regular patterns. We call a regular equation *ordered* if the order in which the variables occur in both sides of the equation is the same; that is, if x and y are variables occurring both in U and V, then x occurs before y in U if and only if x occurs before y in V. Moreover, we say a regular-ordered equation is *strict* if each variable occurs in both sides. For instance $x_1\mathtt{a}x_2x_3\mathtt{b} = x_1\mathtt{a}x_2\mathtt{b}x_3$ is strictly regular-ordered while $x_1\mathtt{a} = x_1x_2$ is regular-ordered (but not strictly since x_2 occurs only on one side) and $x_1\mathtt{a}x_3x_2\mathtt{b} = x_1\mathtt{a}x_2\mathtt{b}x_3$ is regular but not regular-ordered.

In Sect. 3.3 we also consider equations with regular and linear length constraints defined as follows. Given a word equation $U = V$, a set of *linear length constraints* is a system θ of linear Diophantine equations where the unknowns correspond to the lengths of possible substitutions of each variable $x \in X$. Moreover, given a variable $x \in X$, a *regular constraint* is, in this paper, a regular language L_x given by a finite automaton; more general types of regular constraints, imposing that the image of a variable belongs to more than one language, are sometimes used (see [8] and the references therein). The satisfiability of word equations with linear length and/or regular constraints is the question of whether a solution h exists satisfying the system θ and/or such that $h(x) \in L_x$ for each $x \in X$.

3 Results

3.1 Undecidability Results

In this section, we show the undecidability of various extensions of the existential theory of word equations, defined as binary and 3-ary relations which may easily be interpreted as predicates. In each case, undecidability is ultimately obtained by showing that, for a unary-style encoding of integers following [6] (where a number is represented using the length of a string in the form a^*b, so ε is 0, b is 1, etc.), the additional predicate(s) can be used to define a multiplication predicate $\text{Multiply}(x, y, z)$ which decides for numbers i, j, k encoded in this way (i.e., $x = \mathtt{a}^{i-1}\mathtt{b}, y = \mathtt{a}^{j-1}\mathtt{b}, z = \mathtt{a}^{k-1}\mathtt{b}$), whether $k = ij$. Since a corresponding addition predicate can easily be modelled for this encoding using only word equations, undecidability follows immediately.

Definition 1. *Let* AbelianEq, MorphIm, Projection, Subword $\subset A^* \times A^*$ *and* Shuffle, Insert, Erase $\subset A^* \times A^* \times A^*$ *be the relations given by:*

– $(x, y) \in$ AbelianEq *iff* x *and* y *are abelian-equivalent,*

– $(x, y) \in \text{MorphIm}$ *iff there exists a morphism* $h : A^* \to A^*$ *such that* $h(x) = y$,
– $(x, y) \in \text{Projection}$ *iff there exists a projection* $\pi : A^* \to A^*$ *such that* $\pi(x) = y$,
– $(x, y) \in \text{Subword}$ *iff* x *is a (scattered) subword of* y.
– $(x, y, z) \in \text{Shuffle}$ *iff* $z \in x\Delta y$,
– $(x, y, z) \in \text{Erase}$ *iff* z *is obtained from* x *by removing some occurrences of* y,
– $(x, y, z) \in \text{Insert}$ *iff* z *is obtained from* x *by inserting some occurrences of* y.

For each of the above relations we can also define a predicate with the same name which returns true iff the tuple of arguments belongs to the relation.

The membership problems for all the above relations are in NP, and therefore decidable. Our main result of this section concerns the MorphIm predicate:

Theorem 1. *Let* $|A| \geq 3$. *Then given the predicate* MorphIm*, the predicate* Multiply *is definable by an existential formula.*

Proof. Assume that A contains at least three distinct letters: $\mathtt{a}, \mathtt{b}, \mathtt{c}$. We shall actually define a predicate $\text{Multiply}_2(x, y, z)$ which returns true iff $x = \mathtt{a}^i\mathtt{b}, y = \mathtt{a}^j\mathtt{b}, z = \mathtt{a}^{ij}\mathtt{b}$ and $ij \geq 2$. Note that we can immediately obtain Multiply from this, as $\text{Multiply}(x, y, z) = \text{Multiply}_2(\mathtt{a}x, \mathtt{a}y, \mathtt{a}z)$ for $x, y, z \neq \varepsilon$ (assuming also $x = y = z = \mathtt{b}$ does not hold). The exceptional cases, when $ij < 2$ can easily be handled individually. We define first a predicate checking some 'initial conditions':

$$\begin{aligned} init(x, x', x'', y, y', z, z') := \exists w, w', w''.\, x' = w\mathtt{a} \wedge y' = w'\mathtt{a} \wedge (x' = w''\mathtt{aa} \vee y' = w''\mathtt{aa}) \\ \wedge\, x'\mathtt{a} = \mathtt{a}x' \wedge y'\mathtt{a} = \mathtt{a}y' \wedge z'\mathtt{a} = \mathtt{a}z' \wedge x = x'\mathtt{b} \wedge y = y'\mathtt{b} \wedge z = z'\mathtt{b} \;\wedge x''x = xx''. \end{aligned}$$

Recalling Lemma 1, it is straightforward to see that *init* evaluates to true if and only if there exist $i, j, k, \ell, p \in \mathbb{N}_0$ with $ij \geq 2$ such that (1) $x' = \mathtt{a}^i$, $y' = \mathtt{a}^j$, $z' = \mathtt{a}^k$, and (2) $x = \mathtt{a}^i\mathtt{b}$, $y = \mathtt{a}^j\mathtt{b}$, $z = \mathtt{a}^k\mathtt{b}$, and (3) $x'' = (\mathtt{a}^i\mathtt{b})^p$. Now we give the definition of Multiply_2 as follows:

$$\begin{aligned} \text{Multiply}_2(x, y, z) := \exists x', x'', y', z', u, v.\; init(x, x', x'', y, y', z, z') \wedge \text{MorphIm}(x'', y') \\ \wedge\, \text{MorphIm}(y', x'') \;\wedge \text{MorphIm}(u, v) \wedge u = x''\mathtt{cc}x''x'\mathtt{ccb} \;\wedge v = z'\mathtt{cc}z'x'\mathtt{cc}. \end{aligned}$$

Suppose that Conditions (1)–(3) are met (i.e., *init* is satisfied). Consider the subclause $\text{MorphIm}(x'', y') \wedge \text{MorphIm}(y', x'')$. This is satisfied if and only if there exist morphisms $g, h : A^* \to A^*$ such that $g((\mathtt{a}^i\mathtt{b})^p) = \mathtt{a}^j$ and $h(\mathtt{a}^j) = (\mathtt{a}^i\mathtt{b})^p$. Clearly, the latter implies that p is a multiple of j, while the former implies that j is a multiple of p, and hence if both are satisfied then $j = p$. On the other hand, if $j = p$, then it is easy to construct such morphisms (g maps $\mathtt{b}$ to $\mathtt{a}$ and $\mathtt{a}$ to ε while h maps $\mathtt{a}$ to $\mathtt{a}^i\mathtt{b}$). Thus this subclause is satisfied in addition to the *init* predicate if and only if Conditions (1)–(3) hold for $p = j$. By elementary substitutions, the remaining part (i.e., $\text{MorphIm}(u, v) \wedge u = x''\mathtt{cc}x''x'\mathtt{ccb} \wedge v = z'\mathtt{cc}z'x'\mathtt{cc}$) is also satisfied if and only if $u = (\mathtt{a}^i\mathtt{b})^j\mathtt{cc}(\mathtt{a}^i\mathtt{b})^j\mathtt{a}^i\mathtt{ccb}$, and $v = (\mathtt{a}^k\mathtt{cca}^{k+i}\mathtt{cc})$. It remains to show that there exists a morphism $f : A^* \to A^*$ such that $f(u) = v$ if and only if $k = ij$. In the case that $k = ij$, the morphism f may be given e.g. by $f(\mathtt{a}) = \mathtt{a}$, $f(\mathtt{b}) = \varepsilon$ and $f(\mathtt{c}) = \mathtt{c}$. For the other direction, assume that such a morphism f exists. Firstly, consider the case that $f(\mathtt{c}) \in \{\mathtt{a}, \mathtt{b}\}^*$. Then $\mathtt{c}$ must occur in $f(\mathtt{a})$ or $f(\mathtt{b})$. However, under our assumption that $ij \geq 2$, this implies $|f(u)|_{\mathtt{c}} > 4$ meaning $f(u) \neq v$ which is a contradiction. Consequently, we may infer that $f(\mathtt{c})$ contains the letter $\mathtt{c}$. Then since $|u|_{\mathtt{c}} = |v|_{\mathtt{c}}$, it follows that $f(\mathtt{c}) = v_1 \mathtt{c} v_2$ where $v_1, v_2 \in \{\mathtt{a}, \mathtt{b}\}^*$. Thus $f(u) = f((\mathtt{a}^i\mathtt{b})^j) v_1 \mathtt{c} v_2 v_1 \mathtt{c} v_2 f((\mathtt{a}^i\mathtt{b})^j) \mathtt{a}^i v_1 \mathtt{c} v_2 v_1 \mathtt{c} v_2 f(\mathtt{b})$. It follows that $v_1 = v_2 = \varepsilon$, and thus that $f(\mathtt{b}) = \varepsilon$. Hence, $f(\mathtt{a}^{ij}) = \mathtt{a}^k$ and $f(\mathtt{a}^{ij+i}) = \mathtt{a}^{k+i}$ must hold. Clearly, $f(\mathtt{a}) = \mathtt{a}^n$ for some $n \in \mathbb{N}$. Thus we have $nij = k$ and $nij + ni = k + i$. Hence, $n = 1$ and $k = ij$, as required. □

Unlike for the other predicates below, our construction for MorphIm relies strictly on the alphabet A having at least three letters. This is in particular contrast to many other results on the (un)decidability theories of word equations which are usually independent of alphabet size (provided $|A| \neq 1$). Thus we consider it to be of particular interest to settle the remaining open case of whether Theorem 1 holds also for binary alphabets A.

As previously mentioned, further to the predicate MorphIm, many other natural predicates dealing with basic properties and relationships of words lead to undecidability. The following result concerns the remaining predicates listed in Definition 1, and it is obtained by reducing to predicates $Only\mathtt{a}s(x, y)$ and $Only\mathtt{b}s(x, y)$ which return true if and only if $y = \mathtt{a}^{|x|_\mathtt{a}}$ (respectively, $y = \mathtt{b}^{|x|_\mathtt{b}}$). Büchi and Senger [6] show how these predicates can easily be used to model multiplication, and thus undecidability follows.

Proposition 1. *Given any of the predicates* AbelianEq*,* Shuffle*,* Projection*,* Subword*,* Insert*,* Erase*, the predicates Only*$\mathtt{a}$*s and Only*$\mathtt{b}$*s are definable by existential formulas.*

The next theorem sums up the consequences of Proposition 1 and Theorem 1.

Theorem 2. *The existential theory of word equations becomes undecidable when augmented with any of the following predicates:* AbelianEq*,* Shuffle*,* Projection*,* Subword*,* MorphIm *(if $|A| \geq 3$),* Insert*,* Erase*.*

3.2 Quantifier Alternation

Next, we focus on extending the existential theory of word equations by allowing, instead of new predicates, quantifier alternation.

Firstly, recall the *Inclusion of Pattern Languages* problem (IPL, for short, see [5,17]): given two patterns $\alpha \in (A \cup X)^*$ and $\beta \in (A \cup Y)^*$, where A is an alphabet of constants with at least two distinct letters and X and Y are disjoint sets of variables, decide whether $L(\alpha) \subseteq L(\beta)$. IPL admits a reformulation in terms of word equations: decide whether the formula $\exists x_1, \ldots, x_n . \forall y_1, \ldots, y_m . \alpha \neq \beta$ holds in A^*. As IPL is undecidable for terminal alphabets of size 2 or more [5,11], it immediately follows that checking the truth value of $\exists^*\forall^*$-quantified inequation $U \neq V$ in A^*, with $|A| \geq 2$, is undecidable even when

U and V do not contain any common variable, as long as the number of terminals occurring in UV is at least two. This exhibits a very simple fragment of Σ_2 that is undecidable.

Further, we show two normal form results for the Σ_2-fragment of $\mathsf{FO}(A^*, \cdot)$.

Proposition 2. *Let $A \neq \emptyset$ be an alphabet. For every formula ϕ in the Σ_2-fragment of $FO(A^*, \cdot)$ we can construct a Σ_2 terminal-free formula ψ, which holds in A^* iff ϕ holds in A^*.*

Proposition 3. *Let A be an alphabet, $|A| \geq 2$. For every formula ϕ in the Σ_2-fragment of $FO(A^*, \cdot)$ we can construct $\psi = \exists x_1, \ldots, x_n.\forall y_1, \ldots, y_m.U \neq V$, with $U, V \in (A \cup \{x_1, \ldots, x_n, y_1, \ldots, y_m\})^*$, such that ϕ holds in A^* if and only if ψ holds in A^*.*

Note that Proposition 3 does not follow directly by applying the results of [18] to the initial arbitrary formula, in order to reduce it to a single equation. This would have lead to an $\exists^*\forall^*\exists^*$-quantified positive equation, so not to a Σ_2-formula.

The results in Propositions 2 and 3 as well as the remarks regarding IPL show that it is undecidable to check whether some very simple formulae hold in A^*, when $|A| \geq 2$. Also, it is worth noting that applying first Proposition 2 and then Proposition 3 to an arbitrary Σ_2-formula would lead to a single $\exists^*\forall^*$-quantified inequation which contains two terminals, as the constructions in [18] (used in the proof of Proposition 3) require at least two terminals in the equation. However, unlike the inequations encoding IPL instances, the one we obtain by applying our two propositions does not necessarily fulfil the condition that its sides are variable disjoint. Thus, it is natural to ask whether every Σ_2-formula can be reduced to an inequation encoding an instance of IPL. We conjecture that the answer to this question is no.

We have showed that deciding whether a Σ_2-formula, whose sides contain two terminals, holds in A^* for some $|A| \geq 2$ is undecidable. It is possible to show that, when $|A| \geq 2$, for every word equation (which can encode any formula from the Σ_1-fragment of $FO(A^*, \cdot)$, by [18]) we can construct a word equation whose sides contain exactly two terminals $\mathtt{a}$ and $\mathtt{b}$, and whose solutions over $\{\mathtt{a}, \mathtt{b}\}$ bijectively correspond to the solutions of the initial equation. Thus, solving a word equation whose sides contain two terminals is as complex as solving arbitrary word equations.

Hence, we will investigate next which is the case of Σ_1 and Σ_2-formulae over word equations whose sides contain at most one terminal. Proposition 2 already gives us a first answer: checking whether a Σ_2-terminal-free formula holds in A^*, with $|A| \geq 2$, is undecidable. On the other hand, checking whether a formula from $FO(A^*, \cdot)$, whose sides contain at most one terminal $\mathtt{a}$, holds in $\{\mathtt{a}\}^*$ is decidable, as it can be canonically seen as a formula in the Presburger arithmetic.

We concentrate now on other decidable variants. In all these cases, we augment our signature with linear arithmetic over the lengths of variables; all decidability results obtained in this setting hold canonically for the case when such restrictions do not appear. We first look at equations without any quantifier alternation.

Proposition 4. *Let* $\mathtt{a} \in A$*. The satisfiability in* A^* *of quantifier-free positive formulae over word equations* $U = V$*, with* $U, V \in (X \cup \{\mathtt{a}\})^*$*, with linear length constraints is NP-complete.*

Complementing the above result, we show that the satisfiability of quantifier-free first order formulae over word equations $U = V$ (so including negation), such that $U, V \in (X \cup \{\mathtt{a}\})^*$, with linear length constraints is equivalent to solving arbitrary word equations with length constraints. Hence, at the moment, we cannot say anything about the decidability of such formulae. One direction of our result is immediate, while the other follows similarly to Proposition 2.

Theorem 3. *Let* $|A| \geq 2$ *and* $\mathtt{a} \in A$*. Given an equation* $U = V$*, with* $U, V \in (A \cup X)^*$*, with linear length constraints* θ*, there exists a system* $\mathcal{S}$ *of positive and negative equations* $U_i = V_i$ *or* $U_i \neq V_i$ *with* $U_i, V_i \in (X' \cup \{\mathtt{a}\})^*$ *and* $X \subset X'$ *with linear length constraints* θ'*, such that* $\mathcal{S}$ *is satisfiable (in* A^**) if and only if* $U = V$ *is satisfiable.*

Building on Proposition 4, Theorem 4 considers the Σ_2 fragment in the case that only one terminal letter may appear in the equations. Note that this does not necessarily imply $|A| = 1$. If the positive theory only is considered, augmented with the Length predicate defined in the previous section (i.e., Length(x, y) is true if and only if $|x| = |y|$), then we obtain a decidable fragment. Note in particular that the Length predicate can be used in conjunction with simple equations to model arbitrary linear length constraints.

Theorem 4. *Let* $\mathtt{a} \in A$*. The positive* Σ_2*-fragment, restricted to word equations containing only the terminal symbol* $\mathtt{a}$*, augmented with* Length*, is decidable.*

Firstly, we need the following lemma. Then, we give the full proof of Theorem 4.

Lemma 3. *Let* $Y = \{y_1, y_2, \ldots, y_n\} \subseteq X$ *and let* $U, V \in (Y \cup A)^*$*. Let* $k > |UV|$ *and let* $h : X^* \to A^*$ *be the substitution such that* $h(y_i) = \mathtt{a}\mathtt{b}^{k+i}\mathtt{a}$*. Then* $h(U) = h(V)$ *if and only if* $U = V$ *(the strings* U *and* V *coincide).*

Proof. (Theorem 4) W. l. o. g. we may assume that all arguments of the Length predicate are either single variables or words in A^*. Indeed, if we have a "longer" argument α over $(X \cup A)^*$, we can replace it with a new variable x and add the equation $x = \alpha$. For the purposes of this proof we shall say that a term is trivial if, for all the word equations $U = V$, U and V are identical, and moreover, all Length predicates of the form Length(x, y) where either $x = y \in X$ or $x, y \in A^*$ and $|x| = |y|$. If $|A| = 1$, decidability follows from the decidability of Presburger arithmetic. Thus we may assume $\mathtt{a}, \mathtt{b} \in A$ with $\mathtt{a} \neq \mathtt{b}$. W.l.o.g. we may assume that we have a sentence in disjunctive normal form as follows:

$$\exists x_1, x_2, \ldots, x_n . \forall y_1, y_2, \ldots, y_m . (e_{1,1} \wedge \ldots \wedge e_{1,k_1}) \vee \ldots \vee (e_{t,1} \wedge \ldots \wedge e_{t,k_t}), \quad (1)$$

where the $e_{i,j}$ are either: (1) of the form Length(z_1, z_2) where z_1 and z_2 are in $\{x_1, \ldots, x_n, y_1, \ldots, y_m\} \cup A^*$, or (2) individual word equations over the variables $x_1, \ldots, x_n, y_1, \ldots, y_m$ and the terminal $\mathtt{a}$.

We shall show that an assignment for $x_1, x_2, \ldots, x_n$ satisfies (1) if and only if there exists $s, 1 \leq s \leq t$ such that all the resulting atoms $e_{s,i}$ become trivial. The 'if' direction is straightforward, thus we consider the 'only if' direction. Suppose the $x_1, x_2, \ldots, x_n$ are fixed, and consider the result of each $e_{i,j}$ under the substitution. Suppose that for each $s, 1 \leq s \leq t$ there exists $r_s, 1 \leq r_s \leq k_s$ such that e_{s,r_s} is non-trivial. Let p be the maximum over the lengths of all constant terms in the sentence, lengths of the x_i, and lengths of equations given by the type-(2) atoms $e_{i,j}$ for $1 \leq i \leq t, 1 \leq j \leq k_i$. Consider the choice of $y_1, y_2, \ldots, y_m$ given by $y_k = \mathtt{ab}^{p+k}\mathtt{a}$ for $1 \leq k \leq m$. By Lemma 3, if e_{s,r_s} is of type (2), then it will evaluate to false. If e_{s,r_s} is of type (1), then we have three cases. Firstly, if both arguments to the Length predicate are constant terms in A^*, then clearly e_{s,r_s} will evaluate to false since it is non-trivial. Similarly, since the y_i are longer than all constant terms and substituted values of the x_ks, if exactly one of the arguments is a constant in A^* while the other is a variable in $\{y_1, y_2, \ldots, y_m\}$, then e_{s,r_s} will also evaluate to false. Finally, since $|y_\ell| \neq |y'_\ell|$ for all $\ell \neq \ell'$, if both arguments are variables, e_{s,r_s} will again evaluate to false. Summarising the above, for any given choice of $x_1, x_2, \ldots, x_n$ there exists a choice of $y_1, y_2, \ldots, y_m$ such that any of the conjunctions containing a non-trivial equation or Length predicate will be false. It follows that the sentence is satisfiable if and only if there exists a choice for $x_1, x_2, \ldots, x_n$ and $s, 1 \leq s \leq t$ such that all the $e_{s,i}$ terms, $1 \leq i \leq k_s$ become trivial.

For terms $e_{i,j}$ of type (2), this is reduced to solving a system of existentially quantified word equations over $x_1, x_2, \ldots, x_n$ as follows: suppose $e_{i,j}$ is the equation $u_0 y_{i_1} u_1 y_{i_2} u_2 \ldots y_{i_p} u_p = v_0 y_{j_1} v_1 y_{j_2} \ldots y_{j_q} v_q$, where $p, q \in \mathbb{N}_0$, $i_k, j_\ell \in [1, m]$ for $1 \leq k \leq p$ and $1 \leq \ell \leq q$, and $u_k, v_\ell \in (\{x_1, x_2, \ldots, x_n\} \cup A)^*$ for $1 \leq k \leq p$ and $1 \leq \ell \leq q$. Clearly, for a given choice of values for $x_1, \ldots, x_n$, the equation $e_{i,j}$ becomes trivial if and only if $p = q$, and $u_0 = v_0, u_1 = v_1, \ldots, u_p = v_p$, that is, if $x_1, \ldots, x_n$ forms a solution to the system of equations $u_0 = v_0, u_1 = v_1, \ldots, u_p = v_p$ over the variables $x_1, \ldots, x_n$ and terminal symbols from A.

For a term $e_{i,j}$ of type (1), observe that they may only become trivial under some substitution for the x_ℓs either if it is already trivial, in which case it can just be removed, or if both arguments are in $\{x_1, x_2, \ldots, x_n\}$. Thus, any of the clauses $(e_{i,1} \wedge \ldots \wedge e_{i,k_i})$ containing a term $e_{i,j}$ not conforming to these two cases can be removed entirely. After these two phases of removal, it remains to solve, for each s, $1 \leq s \leq t$, a system of equations (i.e., the conjunctions of the systems derived from the $e_{s,j}$ terms of type (2), as described above) subject to a system of linear length constraints (derived from the terms of type (1)). The resulting equations will also only contain the terminal symbol $\mathtt{a}$, since they are taken directly from the original equations, so the decidability follows from Proposition 4. ◻

Note that the reasoning above can be modified in a straightforward way to get decidability of the positive Σ_2 fragment in the general case (but without length constraints), by substituting any of the well-known algorithms for solving existentially quantified systems of equations (e.g. Makanin's algorithm, Plandowski's algorithm, Recompression) in place of Proposition 4. The resulting proof has similar arguments to those of [9,23], although these results do not address this case directly. Also, the decidability result shown in Theorem 4 is, in a sense, optimal, as checking the truth of terminal-free arbitrary Σ_2-formulae is undecidable.

Corollary 1. *The truth of* Σ_2^+*-formulae over* A^* *is decidable.*

3.3 Decidability with Restricted Form

Following the results of the previous section, we explore one more decidable fragment of $FO(A^*, \cdot)$. More precisely, instead of restricting the terminal symbols appearing in the equation(s) we restrict the variables, considering one of the simplest cases of equations that are not in solved form, thus right at the border of the equations that can be solved by practical string solvers [12,32]. We are able to obtain decidability when augmenting the theory simultaneously with linear arithmetic over variable lengths, regular constraints given as DFAs, as well as constraints based on the predicates Subword and $Eq_{\mathtt{a}}$ from the previous section. Formally, we say that subword (resp. $Eq_{\mathtt{a}}$, abelian) constraints are sets of pairs of variables $(x, y) \in X^2$. Solving equation with these constraints requires asserting that for each such pair, the corresponding predicate returns true (so that, e.g., for each abelian constraint (x, y), the substitutions for x and y are abelian equivalent).

Theorem 5. *The problem of solving strictly regular ordered equations with regular constraints given by DFAs, linear length constraints,* $Eq_{\mathtt{a}}$ *constraints (for each* $\mathtt{a} \in A$*), abelian constraints, and subword constraints is NP-complete.*

Proof. Here we present only a sketch of the proof of this theorem. The proof rests on the fact that solutions to strictly regular ordered equations have a particularly well-suited form for parameterisation. In particular, by applying some canonical arguments from the field of combinatorics on words, it can be shown that the set of solutions is spanned by parametric solutions of the form $h(x) = (u_x v_x)^{n_x} u_x$ where $|u_x v_x|$ is linear in the length of the equation, and n_x may be any positive integer if x is "overlapping" (i.e., some part of $h(x)$ on the LHS coincides with part of $h(x)$ on the RHS) and 0 otherwise. Thus, when deciding if a solution exists to the equation which also satisfies the length and regular constraints, it is sufficient to firstly guess such a parametric form, and then decide whether there exist values for the parameters n_x such that the additional constraints are satisfied. Deciding which values of the parameters are also valid under the regular constraints can be done in an efficient way (non-deterministically) due to Lemma 4: simply guess them. Subword constraints are handled in the same way due to a similar technical result, Lemma 5.

Lemma 4. *Let* L *be a regular language given by a DFA,* M*, with* n *states. Let* $u, v \in A^*$*. Then there exist* $q \in \mathbb{N}_{\leq n}$*,* $P, S \subseteq \mathbb{N}_{\leq n} \cup \{0\}$ *such that the intersection of* $(uv)^+u$ *and* L *is given by* $\{(uv)^s u \mid s \in S\} \cup \{(uv)^{q\mu+p} u \mid \mu \in \mathbb{N} \wedge p \in P\}$.

Lemma 5. *Let* $u, v, u', v' \in A^*$. *Let* $S = \{(p, q) \mid (uv)^p u$ *is a subword of* $(u'v')^q u'\}$. *Then either* $S = \emptyset$, *or there exist integers* $p_1, p_2, q_1, q_2, q_{3,0}, \ldots, q_{3,p_2-1}$, *and* $r_1, r_2, \ldots, r_{p_1+p_2-1}$, *all polynomial in* $|uvu'v'|$, *such that*

$$S = S' \cup \bigcup_{1 \leq i < p_2} \{(p, q) \mid p = p_1 + kp_2 + i \wedge q \geq q_1 + kq_2 + q_{3,i} \wedge k \in \mathbb{N}\}$$

where $S' = \{(p, q) \mid p < p_1 + p_2 \wedge q \geq r_p\}$. *Moreover, this representation of* S *can be computed in (nondeterministic) polynomial time.*

Having so-far obtained expressions for parametric solutions satisfying the equation and the regular constraints and subword constraints, it remains to check whether any of the remaining possibilities also satisfy the length, abelian, and $Eq_{\mathtt{a}}$ constraints. It is straightforward, having already guessed the values u_x and v_x, to convert the latter two into length constraints. Thus finding solutions satisfying all constraints is eventually reduced to solving a linear system of Diophantine equations where the unknowns are the parameters. Since the resulting coefficients can be shown to be at most exponentially large, this is possible in non-deterministic polynomial time, see [26]. □

For regular-ordered equations without the strictness (i.e. variables may occur in only one side), the equivalent of Theorem 5 does not hold. It is a straightforward exercise that regular-ordered equations where each side has only one singly-occurring variable, with regular constraints given by DFAs, is PSPACE-complete. This follows from the fact that determining whether the intersection of n DFAs is empty is PSPACE-hard. Similarly, the undecidability proofs for the predicates described in Sect. 3.1 require only very restricted combinations of equations, so we can expect that when such constraints are added, strict restrictions on the structure are necessary for maintaining decidability.

References

1. Abdulla, P.A., Atig, M.F., Chen, Y., Holík, L., Rezine, A., Rümmer, P., Stenman, J.: Norn: an SMT solver for string constraints. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9206, pp. 462–469. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21690-4_29
2. Aydin, A., Bang, L., Bultan, T.: Automata-based model counting for string constraints. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9206, pp. 255–272. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21690-4_15
3. Barrett, C., Conway, C.L., Deters, M., Hadarean, L., Jovanović, D., King, T., Reynolds, A., Tinelli, C.: CVC4. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 171–177. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_14
4. Berzish, M., Ganesh, V., Zheng, Y.: Z3str3: a string solver with theory-aware heuristics. In: Proceedings of the FMCAD 2017, pp. 55–59. IEEE (2017)
5. Bremer, J., Freydenberger, D.D.: Inclusion problems for patterns with a bounded number of variables. Inf. Comput. **220**, 15–43 (2012)

6. Büchi, J.R., Senger, S.: Definability in the existential theory of concatenation and undecidable extensions of this theory. Z. für math. Logik Grundlagen d. Math. **47**, 337–342 (1988)
7. Day, J.D., Manea, F., Nowotka, D.: The hardness of solving simple word equations. In: Proceedings of the MFCS 2017. LIPIcs, vol. 83, pp. 18:1–18:14 (2017)
8. Diekert, V., Jeż, A., Plandowski, W.: Finding all solutions of equations in free groups and monoids with involution. Inf. Comput. **251**, 263–286 (2016)
9. Diekert, V., Lohrey, M.: Existential and positive theories of equations in graph products. Theory Comput. Syst. **37**(1), 133–156 (2004)
10. Durnev, V.G.: Undecidability of the positive ∀∃-theory of a free semigroup. Sib. Math. J. **36**(5), 917–929 (1995)
11. Freydenberger, D.D., Reidenbach, D.: Bad news on decision problems for patterns. Inf. Comput. **208**(1), 83–96 (2010)
12. Ganesh, V., Minnes, M., Solar-Lezama, A., Rinard, M.: Word equations with length constraints: what's decidable? In: Biere, A., Nahir, A., Vos, T. (eds.) HVC 2012. LNCS, vol. 7857, pp. 209–226. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39611-3_21
13. Halfon, S., Schnoebelen, P., Zetzsche, G.: Decidability, complexity, and expressiveness of first-order logic over the subword ordering. In: Proceedings of the LICS 2017, pp. 1–12. IEEE Computer Society (2017)
14. Hilbert, D.: Mathematische probleme. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse **1900**, 253–297 (1900)
15. Jeż, A.: Recompression: a simple and powerful technique for word equations. In: Proceedings of the STACS 2013. LIPIcs, vol. 20, pp. 233–244 (2013)
16. Jeż, A.: Word equations in nondeterministic linear space. In: Proceedings of the ICALP 2017. LIPIcs, vol. 80, pp. 95:1–95:13 (2017)
17. Jiang, T., Salomaa, A., Salomaa, K., Yu, S.: Decision problems for patterns. J. Comput. Syst. Sci. **50**(1), 53–63 (1995)
18. Karhumäki, J., Mignosi, F., Plandowski, W.: The expressibility of languages and relations by word equations. J. ACM (JACM) **47**(3), 483–505 (2000)
19. Kiezun, A., Ganesh, V., Guo, P.J., Hooimeijer, P., Ernst, M.D.: HAMPI: a solver for string constraints. In: Proceedings of the ISSTA 2009, pp. 105–116. ACM (2009)
20. Lin, A.W., Barceló, P.: String solving with word equations and transducers: towards a logic for analysing mutation XSS. In: ACM SIGPLAN Notices. vol. 51, pp. 123–136. ACM (2016)
21. Lothaire, M.: Combinatorics on Words. Addison-Wesley, Boston (1983)
22. Makanin, G.S.: The problem of solvability of equations in a free semigroup. Sb.: Math. **32**(2), 129–198 (1977)
23. Makanin, G.S.: Decidability of the universal and positive theories of a free group. Math. USSR-Izv. **25**(1), 75 (1985)
24. Manea, F., Nowotka, D., Schmid, M.L.: On the solvability problem for restricted classes of word equations. In: Brlek, S., Reutenauer, C. (eds.) DLT 2016. LNCS, vol. 9840, pp. 306–318. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53132-7_25
25. Matiyasevich, Y.V.: A connection between systems of words-and-lengths equations and hilbert's tenth problem. Zapiski Nauchnykh Seminarov POMI **8**, 132–144 (1968)
26. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM (JACM) **28**(4), 765–768 (1981)

27. Plandowski, W.: Satisfiability of word equations with constants is in PSPACE. In: Proceedings of the FOCS 1999, pp. 495–500. IEEE (1999)
28. Quine, W.V.: Concatenation as a basis for arithmetic. J. Symb. Log. **11**(4), 105–114 (1946)
29. Trinh, M.-T., Chu, D.-H., Jaffar, J.: Progressive reasoning over recursively-defined strings. In: Chaudhuri, S., Farzan, A. (eds.) CAV 2016. LNCS, vol. 9779, pp. 218–240. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41528-4_12
30. Vazenin, J.M., Rozenblat, B.V.: Decidability of the positive theory of a free countably generated semigroup. Math. USSR Sb. **44**(1), 109–116 (1983)
31. Yu, F., Alkhalaf, M., Bultan, T.: Stranger: an automata-based string analysis tool for PHP. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 154–157. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12002-2_13
32. Zheng, Y., Ganesh, V., Subramanian, S., Tripp, O., Berzish, M., Dolby, J., Zhang, X.: Z3str2: an efficient solver for strings, regular expressions, and length constraints. Form. Methods Syst. Des. **50**(2–3), 249–288 (2017)