Hardness Results for Constant-Free Pattern Languages and Word Equations

Aleksi Saarela
Department of Mathematics and Statistics, University of Turku, Finland
amsaar@utu.fi

Abstract

We study constant-free versions of the inclusion problem of pattern languages and the satisfiability problem of word equations. The inclusion problem of pattern languages is known to be undecidable for both erasing and nonerasing pattern languages, but decidable for constant-free erasing pattern languages. We prove that it is undecidable for constant-free nonerasing pattern languages. The satisfiability problem of word equations is known to be in PSPACE and NP-hard. We prove that the nonperiodic satisfiability problem of constant-free word equations is NP-hard. Additionally, we prove a polynomial-time reduction from the satisfiability problem of word equations to the problem of deciding whether a given constant-free equation has a solution morphism α such that $\alpha(xy) \neq \alpha(yx)$ for given variables x and y.

2012 ACM Subject Classification Mathematics of computing → Combinatorics on words

Keywords and phrases Combinatorics on words, pattern language, word equation

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.140

Category Track B: Automata, Logic, Semantics, and Theory of Programming

1 Introduction

The first topic of this article is pattern languages. If we fix an alphabet of variables and an alphabet of constants, we can define a pattern as a word consisting of variables and constant letters (constants are often called terminals). Given a pattern U, the nonerasing pattern language of U is the set of images of U under all morphism that map the variables to nonempty constant words and preserve the constants. The erasing pattern language of U is defined in a similar way, except that the variables can be mapped also to the empty word. Nonerasing pattern languages were introduced by Angluin [1] and erasing pattern languages by Shinohara [33].

There are many interesting algorithmic questions about pattern languages, and they are related to applications such as pattern matching and inductive inference. The membership problem of pattern languages, which can also be called the matching problem, is NP-complete in both the nonerasing and erasing case, and so are many of its variations, see, e.g., [10] and [24]. Checking the emptiness of the intersection of two pattern languages is essentially a special case of the satisfiability problem of word equations (discussed later in the introduction), and can therefore be done in polynomial space.

The equivalence problem of two pattern languages is almost trivially decidable in the nonerasing case: If the alphabet of constants is not unary, the nonerasing pattern languages of two patterns are the same if and only if the patterns are identical up to a renaming of the variables [1] (if the alphabet of constants is unary, the problem is a bit more complicated but still easily decidable). In the erasing case, however, the decidability of the equivalence problem is an open question.

The inclusion problem of pattern languages was mentioned as an open question in [1]. It was proved to be undecidable in both the nonerasing and erasing case by Jiang, Salomaa, Salomaa and Yu [18]. They reduced the undecidable problem of determining whether a

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 140; pp. 140:1–140:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


nondeterministic two-counter automaton without input has an accepting computation to the inclusion problem of erasing pattern languages, which they then reduced to the inclusion problem of nonerasing pattern languages. Both proofs are very complicated. They also proved that the inclusion problem of constant-free erasing pattern languages is decidable. This proof is simpler but not trivial.

Analyzing the equivalence and inclusion of pattern languages naturally leads to eight decision problems depending on whether we consider the nonerasing or erasing case and whether we consider patterns with or without constants. By the results mentioned above, the decidability status of six of these eight problems was known after [18], but two questions were left open: Is the equivalence problem decidable for erasing pattern languages, and is the inclusion problem decidable for constant-free nonerasing pattern languages? In [18], a positive answer was conjectured for the first question. The second one was only stated as an open problem with no conjecture. The questions have remained open since then.

The results in [18] were very interesting for many reasons. First, they solved a famous problem that had been open for many years. Second, the inclusion problem is important for inductive inference of pattern languages, see, e.g., the articles of Ng and Shinohara [25] and Reidenbach [28]. Third, nonerasing pattern languages became perhaps the first example of a family of formal languages with a trivially decidable equivalence problem but undecidable inclusion problem. Some other families where the equivalence problem is decidable but the inclusion problem is undecidable are the family of languages accepted by finite deterministic multitape automata (decidability of equivalence proved by Harju and Karhumäki [12]) and the family of deterministic context-free languages (decidability of equivalence proved later by Sénizergues [32]), but in these cases the undecidability result is the easier one. It is also interesting that for the equivalence problem of patterns, the nonerasing case is easier, while for the inclusion problem of constant-free patterns, the erasing case is easier.

A lot of further research on the inclusion problem has been done. For example, Freydenberger and Reidenbach [11] proved that the inclusion problem remains undecidable if the size of the alphabet of constants is fixed to be a positive integer k, as long as $k \geq 2$ in the erasing case and $k \geq 4$ in the nonerasing case. Both variants of the inclusion problem are decidable if the alphabet of constants is unary or infinite. Bremer and Freydenberger [4] proved stronger results: Both the erasing and the nonerasing inclusion problem are undecidable even for a fixed number of variables, as long as this number is large enough, and even if the size of the alphabet of constants is two. This result holds also if the second pattern is required to be constant-free.

In this article, we answer one of the open questions by proving that the inclusion problem of constant-free nonerasing pattern languages is undecidable. The result holds even for a fixed number of variables, as long as this number is large enough, and even if the size of the alphabet of constants is two. Among the problems we have discussed, our new result provides the first example where the constant-free version of a problem is undecidable, and the first example where the decidability status of the nonerasing and erasing version has been proved to be different. See Table 1 for a summary.

Let us now move to the topic of equations. A word equation can be defined as a pair (U, V) of patterns, and a solution can be defined as a constant-preserving morphism α such that $\alpha(U) = \alpha(V)$. Like in the case of pattern languages, there are a couple of variations of word equations. First, we can study either the general case of equations with constants, or the restricted case of constant-free equations. Algorithmic questions are usually studied for equations with constants. Some other questions, such as independence [13, 26] and parameterizability [14, 30], are more often studied for constant-free equations. Second, we

Table 1 The decidability of the equivalence (=) and inclusion (⊆) problem of nonerasing (NE) and erasing (E) languages of constant-free patterns (CF) and patterns with constants (C). The decidable problems have been marked with a plus and the undecidable ones with a minus. The new result proved in this article has been circled.

=	NE	E
CF	+	+
C	+	?

⊆	NE	E
CF	⊖	+
C	−	−

can either allow a solution α to be erasing or require it to be nonerasing. This does not usually make as big of a difference as in the case of the equivalence and inclusion of pattern languages, but it can have an effect on some things, for example, the size of largest known independent systems of equations [20]. We allow solutions to be erasing in this article.

The satisfiability problem of word equations, that is, the problem of deciding whether a given word equation (or a system of equations) has a solution, is one of the major algorithmic problems on words. The satisfiability problem was proved to be decidable by Makanin [23]. A survey of Makanin's algorithm can be found in [8]. The first PSPACE algorithm was given by Plandowski [27]. Jeż gave a simpler PSPACE algorithm [16] and proved that the satisfiability problem is in NSPACE(n) [17]. Linear integer programming and the membership problem of pattern languages can both be easily reduced to the satisfiability problem, so it is NP-hard. The NP-completeness of the satisfiability problem is a big open question.

Many special cases have been analyzed. For one-variable equations, the satisfiability problem can be solved in linear time, as proved by Jeż [15], for two-variable equations, in time $O(n^5)$, as proved by Dąbrowski and Plandowski [7], and for quadratic equations, it is NP-hard, as proved by Robson and Diekert [29]. Some other results can be found in the article of Day, Manea and Nowotka [6]. Some quite powerful generalizations of word equations were proved to be solvable in polynomial space by Diekert and Elder [9].

There is not much research about the satisfiability of constant-free word equations. Constant-free equations always have solutions (at least the one mapping all variables to the empty word, and usually infinitely many other trivial ones), so the natural decision problem for them is to ask whether an equation has a nontrivial solution, for some definition of "nontrivial". It is known that deciding whether a constant-free three-variable equation has a nonperiodic solution is in NP [30]. Nontrivial constant-free equations on one or two variables have only periodic solutions.

Constant-free equations might seem much simpler than general ones, but we prove that deciding whether a given constant-free equation has a nonperiodic solution is NP-hard, and for a given constant-free equation and given variables x, y, deciding whether there exists a solution α such that $\alpha(xy) \neq \alpha(yx)$ is as hard as the general version of the satisfiability problem.

Our proofs are based on the idea of simulating constants by variables in a certain way: We replace the constants by words consisting of new variables, and then we make sure that these words behave sufficiently much like constants by adding prefixes and suffixes to the patterns or new equations to a system of equations. The details differ quite a bit depending on the problem.

2 Preliminaries

First, we recall some standard notation, definitions, and results related to combinatorics on words and free monoids. For more, see [5, 21, 3].

The symbols Σ, Γ, Ξ are always used to denote alphabets. Alphabets are usually finite, but at one point, we use an infinite alphabet. For $k \geq 1$, let $\Sigma_k = \{0, \ldots, k-1\}$. When we need an alphabet of size k, we often use specifically the alphabet Σ_k. The empty word is denoted by ε.

A word U is a *factor* of a word V if there exist words X, Y such that $V = XUY$. If we can choose $X = \varepsilon$, then U is a *prefix*, if we can choose $Y = \varepsilon$, then U is a *suffix*, and if we can choose $X \neq \varepsilon \neq Y$, then U is an *internal factor* of V.

A nonempty word is *primitive* if it is not a power of a shorter word. If $U = V^n$ and V is primitive, then V is a *primitive root* of U.

A language $M \subseteq \Sigma^*$ is a *submonoid* of Σ^* if it is closed under concatenation and contains ε. Let M_1 and M_2 be submonoids of Σ^* and Γ^*. A mapping $\alpha : M_1 \to M_2$ is a *morphism* if $\alpha(UV) = \alpha(U)\alpha(V)$ for all $U, V \in M_1$. The most common case is $M_1 = \Sigma^*$ and $M_2 = \Gamma^*$.

Let $L \subseteq M_1$. A morphism $\alpha : M_1 \to M_2$ is *nonerasing* if $\alpha(x) \neq \varepsilon$ for all $x \neq \varepsilon$, *L-preserving* if $\alpha(x) = x$ for all $x \in L$, *L-periodic* if $\alpha(xy) = \alpha(yx)$ for all $x, y \in L$, and *L-nonperiodic* if $\alpha(xy) \neq \alpha(yx)$ for some $x, y \in L$.

In the following theorem, we have collected some folklore results related to these definitions.

▶ **Theorem 1.** *Let U and V be words.*

1. *$UV = VU$ if and only if there exists a word R such that $U, V \in R^*$.*
2. *Every nonempty word has a unique primitive root.*
3. *If U is primitive, then it is not an internal factor of U^2.*
4. *If $|\Sigma| = 2$, then every Σ-nonperiodic morphism $\Sigma^* \to \Gamma^*$ is injective.*

If every element of a submonoid M of Σ^* has a unique representation as a product of elements of some subset $B \subseteq M$, then M is a *free monoid* and B is its *basis*. Of course, Σ^* is a free monoid and Σ is its basis.

If M is a free monoid with a basis B, then every mapping $\alpha : B \to \Gamma^*$ can be extended to a morphism $M \to \Gamma^*$ in a unique way, and every injective mapping $\alpha : B \to \Gamma$ can be extended to an injective morphism $M \to \Gamma^*$. Moreover, if $L \subseteq B \cap \Gamma$, then every mapping $\alpha : B \setminus L \to \Gamma^*$ can be extended to an L-preserving morphism $M \to \Gamma^*$ in a unique way. Therefore, we often define a morphism α by just saying that it is L-preserving and giving the values $\alpha(x)$ for all $x \in B \setminus L$.

We need the following well-known characterization of free monoids.

▶ **Theorem 2.** *Let M be a submonoid of Σ^*. M is a free monoid if and only if there does not exist words $U, V, W \in \Sigma^*$ such that $U, VW, UV, W \in M$ but $V \notin M$.*

Let Ξ be an alphabet of variables and Σ an alphabet of constants. The alphabets Ξ and Σ are assumed to be disjoint. A *pattern over* (Ξ, Σ) is a word $U \in (\Xi \cup \Sigma)^+$. The pattern U is *constant-free* if $U \in \Xi^+$. The *nonerasing pattern language* of U, denoted by $L_{\mathrm{NE}}(U)$, is the set of images of U under all nonerasing Σ-preserving morphisms $(\Xi \cup \Sigma)^* \to \Sigma^*$.

In the following definitions, by an alphabet of size $\star$, we mean an alphabet of arbitrary finite size. For $m, n, k \in \mathbb{Z}_+ \cup \{\star\}$, we define the following decision problems related to patterns:

- Inclusion problem of nonerasing pattern languages $\mathrm{PatIncl}_{\mathrm{NE}}(m, n, k)$: Given alphabets Ξ_1, Ξ_2, Σ of sizes m, n, k, respectively, a pattern U over (Ξ_1, Σ), and a pattern V over (Ξ_2, Σ), decide whether $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$.
- Inclusion problem of constant-free nonerasing pattern languages $\mathrm{PatIncl}^{\mathrm{CF}}_{\mathrm{NE}}(m, n, k)$: Given alphabets Ξ_1, Ξ_2, Σ of sizes m, n, k, respectively, a constant-free pattern U over (Ξ_1, Σ), and a constant-free pattern V over (Ξ_2, Σ), decide whether $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$.

As mentioned in the introduction, $\mathrm{PatIncl}_{\mathrm{NE}}(\star, \star, \star)$ was shown to be undecidable already in [18]. We need the following stronger result from [4].

▶ **Theorem 3** ([4], Theorem 3.10)**.** $\mathrm{PatIncl}_{\mathrm{NE}}(3, 2554, 2)$ *is undecidable.*

▶ Remark 4. Often, we use symbols from the end of the English alphabet (e.g., u, v, w, x, y, z in the next example) for ordinary variables, symbols from the beginning of the alphabet (e.g., a, b) for special variables that end up playing the role of constants in some sense, and nonnegative integers (e.g., $0, 1$) for the actual constants.

▶ **Example 5.** It is mentioned in several articles [18, 24] that one reason why the inclusion problem of nonerasing pattern languages is so difficult is that many unavoidability properties can be formulated in terms of pattern languages. In our new undecidability proof, an important role is played by one unavoidability result, although an extremely simple one: Every binary word of length at least 4 has a nonempty square factor, and therefore every binary word of length at least 6 has a nonempty internal square factor. By considering patterns over $(\{u, v, w, x, y, z\}, \Sigma_2)$, this can be expressed as $L_{\mathrm{NE}}(uvwxyz) \subseteq L_{\mathrm{NE}}(xy^2z)$. More specifically, we will need the fact that $L_{\mathrm{NE}}(x^2y^2) \setminus L_{\mathrm{NE}}(xy^2z) = \{0011, 1100\}$.

A *word equation over* (Ξ, Σ) is a pair of patterns over (Ξ, Σ). A *solution* of an equation (U, V) is a Σ-preserving morphism $\alpha : (\Xi \cup \Sigma)^* \to \Sigma^*$ such that $\alpha(U) = \alpha(V)$. A *system of equations* is a set of equations. A *solution* of a system is a morphism that is a solution of every equation in the system.

An equation (U, V) is *constant-free* if $U, V \in \Xi^+$. For constant-free equations, Ξ-periodic solutions are considered trivial. We often call these solutions just *periodic* and the others *nonperiodic*.

▶ **Example 6.** Consider word equations over $(\{x, y, z\}, \Sigma_2)$. The equation $(x^2, y0y)$ has no solutions, because $|\alpha(x^2)|$ is even and $|\alpha(y0y)|$ is odd for all Σ_2-preserving morphisms α. The constant-free equation (x^2, yzy) has nonperiodic solutions α defined by

$$\alpha(x) = (PQ)^{i+1}P, \qquad \alpha(y) = (PQ)^iP, \qquad \alpha(z) = QPPQ$$

for all $P, Q \in \Sigma_2^*$, $PQ \neq QP$, $i \in \mathbb{Z}_{\geq 0}$, and periodic solutions α defined by

$$\alpha(x) = P^{i+j}, \qquad \alpha(y) = P^i, \qquad \alpha(z) = P^{2j}$$

for all $P \in \Sigma_2^*$, $i, j \in \mathbb{Z}_{\geq 0}$.

By the theorem of Lyndon and Schützenberger [22], if $\Xi = \{x, y, z\}$ and $k, m, n \geq 2$, then the word equation $(x^k, y^m z^n)$ has only periodic solutions. In other words, if α is a $\{y, z\}$-nonperiodic morphism, then $\alpha(y^m z^n)$ is primitive. In Theorem 7, we state a generalization of this result that we need later. It was proved by Spehner [34] and by Barbin-Le Rest and Le Rest [2]. A shorter proof can be found in [31].

▶ **Theorem 7.** *Let $W \in \{y, z\}^*$ be a primitive word that has at least two occurrences of both letters y and z. Let α be a $\{y, z\}$-nonperiodic morphism. Then $\alpha(W)$ is primitive.*

For $n, k \in \mathbb{Z}_+ \cup \{\star\}$, we define the following decision problems related to systems of word equations:

- Satisfiability problem of word equations $\mathrm{EqSat}(n, k)$: Given alphabets Ξ, Σ of sizes n, k, respectively, and a system S of equations over (Ξ, Σ), decide whether S has a solution.
- Nonperiodic satisfiability problem of constant-free word equations $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(n, k)$: Given alphabets Ξ, Σ of sizes n, k, respectively, and a system S of constant-free equations over (Ξ, Σ), decide whether S has a nonperiodic solution.
- Noncommuting satisfiability problem of constant-free word equations $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}(n, k)$: Given alphabets Ξ, Σ of sizes n, k, respectively, a system S of constant-free equations over (Ξ, Σ), and variables $x, y \in \Xi$, decide whether S has a $\{x, y\}$-nonperiodic solution.

As mentioned in the introduction, $\mathrm{EqSat}(\star,\star)$ is known to be in PSPACE and NP-hard. If Σ is unary, then word equations are essentially linear Diophantine equations, so $\mathrm{EqSat}(\star,1)$ is equivalent to linear integer programming in unary notation, which is known to be NP-complete.

We have defined the above decision problems for systems of equations. Studying single equations instead of systems would not make them much easier, except in the case where Σ is unary. If there are at least two distinct constant letters, then for every finite system of word equations, we can find an equation that has exactly the same solutions as the system, and for every finite system of constant-free word equations, we can find a constant-free equation that has exactly the same nonperiodic solutions as the system, as proved by Hmelevskii [14]. Moreover, these equations can be constructed in polynomial time.

If A and B are decision problems and A is polynomial-time reducible to B, we use the notation $A \leq_p B$. If A and B are polynomially equivalent, that is, $A \leq_p B$ and $B \leq_p A$, then we use the notation $A \equiv_p B$.

3 Inclusion problem of pattern languages

We are going to prove that $\mathrm{PatIncl}_{\mathrm{NE}}(\star,\star,2) \leq_p \mathrm{PatIncl}^{\mathrm{CF}}_{\mathrm{NE}}(\star,\star,2)$. Let the alphabet of constants be Σ_2. Let a and b be new variables that are supposed to represent the constants 0 and 1. Nonerasing morphisms that map a to 0 and b to 1 or vice versa can be called *good*, and other nonerasing morphism can be called *bad*. For all patterns U, V, we must construct constant-free patterns U', V' such that $L_{\mathrm{NE}}(U') \subseteq L_{\mathrm{NE}}(V')$ if and only if $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$. In other words, we must show that the following conditions are satisfied:

1. If $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$, then for all good morphisms α', there exists a nonerasing morphism β' such that $\beta'(V') = \alpha'(U')$.
2. If $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$, then for all bad morphisms α', there exists a nonerasing morphism β' such that $\beta'(V') = \alpha'(U')$.
3. If $L_{\mathrm{NE}}(U) \not\subseteq L_{\mathrm{NE}}(V)$, then there exists a nonerasing morphism α' such that for all nonerasing morphisms β', $\beta'(V') \neq \alpha'(U')$.

Before giving the definition of U' and V' and the formal proofs, we explain some ideas behind the construction.

The simplest idea would be to replace 0 and 1 by a and b. Let U_1, V_1 be the constant-free patterns we get from U, V this way. If we use U_1, V_1 as U', V', then the first condition is satisfied. However, the next example shows that the third condition does not hold in general.

▶ **Example 8.** Let $U = 0x$, $V = 1x$, $U' = ax$, $V' = bx$. Then clearly $L_{\mathrm{NE}}(U) \not\subseteq L_{\mathrm{NE}}(V)$ and $L_{\mathrm{NE}}(U') \subseteq L_{\mathrm{NE}}(V')$, so the third condition does not hold.

The problem with the third condition is that β' does not necessarily map a and b in the same way as α'. To solve this, we use an idea that is somewhat similar to one used in [4, Subsection 5.2], where prefixes are added to patterns to ensure that certain variables must be mapped in a certain way. Consider the patterns

$$U_2 = a^2b^2c^2U_1, \qquad V_2 = a^2b^2c^2V_1,$$

where c is a new variable. If α' is good and $\alpha'(c)$ is a third letter that does not appear in $\alpha'(U_1)$, then it is quite easy to see that $\beta'(V_2) = \alpha'(U_2)$ is possible only if $\beta'(x) = \alpha'(x)$ for all $x \in \{a, b, c\}$. Of course, there is no third letter in Σ_2, but we can define $\alpha'(c) \in \Sigma_2^+$ so that it still acts as a separator in the same way a unique letter would. If we use U_2, V_2 as U', V', then the first and the third condition are satisfied.

Satisfying the second condition without interfering with the third condition is the most difficult part. To solve the problems, consider patterns of the form

$$U_4 = W_1 pq^2 r W_2 a^2 b^2 W_3, \qquad V_4 = W_4 pq^2 r W_5,$$

where p, q, r are new variables and $W_1, \ldots, W_5$ are constant-free patterns. The factors pq^2r and a^2b^2 act as a "switch". If $\beta'(pq^2r) = \alpha'(pq^2r)$, then we can say that the switch is in the first position. If $\beta'(pq^2r) = \alpha'(a^2b^2)$, then we can say that the switch is in the second position. For any α', we can define β' so that $\beta'(pq^2r) = \alpha'(pq^2r)$, so the first position is always possible. On the other hand, we can define β' so that $\beta'(pq^2r) = \alpha'(a^2b^2)$ if and only if α' is bad, so the second position is possible for the bad morphisms but not for the good. This allows us to handle the second condition without causing problems with the other two.

Putting these ideas together leads to the following construction. Let U be a pattern over (Ξ_1, Σ_2) and V a pattern over (Ξ_2, Σ_2). Let a, b, c, p, q, r, s, t be new variables not in $\Xi_1 \cup \Xi_2$. We define a $(\Xi_1 \cup \Xi_2)$-preserving morphism

$$\sigma : (\Xi_1 \cup \Xi_2 \cup \Sigma_2)^* \to (\Xi_1 \cup \Xi_2 \cup \{a, b\})^*, \ \sigma(0) = a, \ \sigma(1) = b.$$

We can construct the constant-free patterns

$$\begin{aligned} U' &= a^2b^2c^2 \cdot c^2 \cdot \sigma(U) \cdot c^2 pq^2 rc \cdot \sigma(V) \cdot c^2 a^2 b^2 c \cdot a \\ V' &= a^2b^2c^2 \cdot sc \cdot \sigma(V) \cdot c^2 pq^2 rc \cdot t \end{aligned} \tag{1}$$

where U' is a pattern over $(\Xi_1 \cup \Xi_2 \cup \{a, b, c, p, q, r\}, \Sigma_2)$ and V' is a pattern over $(\Xi_2 \cup \{a, b, c, p, q, r, s, t\}, \Sigma_2)$.

▶ **Lemma 9.** *If $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$, then $L_{\mathrm{NE}}(U') \subseteq L_{\mathrm{NE}}(V')$,*

Proof. Let $\alpha' : (\Xi_1 \cup \Xi_2 \cup \{a, b, c, p, q, r\})^* \to \Sigma_2^*$ be a nonerasing morphism. We must find a nonerasing morphism $\beta' : (\Xi_2 \cup \{a, b, c, p, q, r, s, t\})^* \to \Sigma_2^*$ such that $\beta'(V') = \alpha'(U')$. There are two cases depending on whether α' is good or bad.

If α' is good, then we can define a nonerasing morphism

$$\alpha : (\Xi_1 \cup \Xi_2 \cup \Sigma_2)^* \to \Sigma_2^*, \ \alpha = \alpha' \circ \sigma \circ \alpha' \circ \sigma.$$

It is easy to check that α is Σ_2-preserving. By the assumption $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$, there exists a nonerasing Σ_2-preserving morphism $\beta : (\Xi_2 \cup \Sigma_2)^* \to \Sigma_2^*$ such that $\beta(V) = \alpha(U)$. We can define a morphism

$$\begin{aligned} \beta' : (\Xi_2 \cup \{a, b, c, p, q, r, s, t\})^* \to \Sigma_2^*, \ &\beta'(x) = \alpha'(\sigma(\beta(x))) \text{ for all } x \in \Xi_2, \\ &\beta'(x) = \alpha'(x) \text{ for all } x \in \{a, b, c, p, q, r\}. \\ &\beta'(s) = \alpha'(c), \\ &\beta'(t) = \alpha'(\sigma(V) c^2 a^2 b^2 ca). \end{aligned}$$

It follows directly from the definition of β' that

$$\begin{aligned} &\beta'(a^2b^2c^2sc) = \alpha'(a^2b^2c^4), \\ &\beta'(c^2pq^2rct) = \alpha'(c^2pq^2rc\sigma(V)c^2a^2b^2ca). \end{aligned} \tag{2}$$

Showing that $\beta'(\sigma(V)) = \alpha'(\sigma(U))$ requires some computations. By using the definition of β' and the fact that σ is Ξ_2-preserving and β is Σ_2-preserving, we get

$$\begin{aligned} &\beta'(\sigma(x)) = \beta'(x) = \alpha'(\sigma(\beta(x))) \text{ for all } x \in \Xi_2, \\ &\beta'(\sigma(x)) = \alpha'(\sigma(x)) = \alpha'(\sigma(\beta(x))) \text{ for all } x \in \Sigma_2. \end{aligned} \tag{3}$$

We have

$$\beta'(\sigma(V)) = \alpha'(\sigma(\beta(V))) = \alpha'(\sigma(\alpha(U))) = \alpha(\alpha'(\sigma(U))) = \alpha'(\sigma(U)), \tag{4}$$

where the first equality follows from $V \in (\Xi_2 \cup \Sigma_2)^+$ and (3), the second from $\beta(V) = \alpha(U)$, the third from $(\alpha' \circ \sigma) \circ \alpha = \alpha \circ (\alpha' \circ \sigma)$, and the fourth from α being Σ_2-preserving. It follows from (2) and (4) that $\beta'(V') = \alpha'(U')$

If α' is bad, then $\alpha'(a^2b^2)$ is either 0^4, 1^4, or a binary word of length at least six. In all cases, it has a nonempty internal factor that is a square, so there exists $P, Q, R \in \Sigma_2^+$ such that $\alpha'(a^2b^2) = PQ^2R$. We can define a morphism

$$\begin{aligned}\beta' : (\Xi_2 \cup \{a,b,c,p,q,r,s,t\})^* \to \Sigma_2^*,\ \beta'(x) &= \alpha'(x) \text{ for all } x \in \Xi_2 \cup \{a,b,c\},\\ \beta'(p) &= P,\\ \beta'(q) &= Q,\\ \beta'(r) &= R,\\ \beta'(s) &= \alpha'(c^2\sigma(U)c^2pq^2r),\\ \beta'(t) &= \alpha'(a).\end{aligned}$$

It follows directly from the definition of β' that $\beta'(V') = \alpha'(U')$. ◀

▶ **Lemma 10.** *If $L_{\mathrm{NE}}(U) \not\subseteq L_{\mathrm{NE}}(V)$, then $L_{\mathrm{NE}}(U') \not\subseteq L_{\mathrm{NE}}(V')$.*

Proof. By the assumption $L_{\mathrm{NE}}(U) \not\subseteq L_{\mathrm{NE}}(V)$, there exist a nonerasing Σ_2-preserving morphism $\alpha : (\Xi_1 \cup \Sigma_2)^* \to \Sigma_2^*$ such that $\beta(V) \neq \alpha(U)$ for all nonerasing Σ_2-preserving morphisms $\beta : (\Xi_2 \cup \Sigma_2)^* \to \Sigma_2^*$. We can define a morphism

$$\begin{aligned}\alpha' : (\Xi_1 \cup \Xi_2 \cup \{a,b,c,p,q,r\})^* \to \Sigma_2^*,\ \alpha'(x) &= \alpha(x) \text{ for all } x \in \Xi_1,\\ \alpha'(a) &= 0,\\ \alpha'(b) &= 1,\\ \alpha'(c) &= 10^N1,\\ \alpha'(x) &= 1 \text{ for all } x \in \Xi_2 \cup \{p,q,r\},\end{aligned}$$

where $N = 1 + \max\{2, |\alpha'(\sigma(U))|, |\alpha'(\sigma(V))|\}$.

It is easy to see that $\alpha'(U')$ does not contain any other occurrences of $\alpha'(c)$ than the ten obvious ones. We can show that if A^2 is a nonempty square prefix of $\alpha'(U')$, then $A = \alpha'(a) = 0$. First, if $\alpha'(a^2b^2c^3)$ is a prefix of A, then $A\alpha'(a^2b^2c^3)$ is a prefix of $\alpha'(U')$, which is impossible, because $\alpha'(c^3)$ does not have any occurrences in $\alpha'(U')$ starting after the prefix $\alpha'(a^2b^2c^3)$. Second, if A is a prefix of $\alpha'(a^2b^2c^3)$ and $|A| \geq 5$, then $A00111$ is a prefix of $\alpha'(a^2b^2c^4)$, which is impossible, because 111 does not have any occurrences in $\alpha'(c^4)$. Finally, if $|A| \leq 4$, then clearly the only possibility is $A = 0$. Similarly, we can show that if W is the word such that $U' = a^2b^2W$, then the only nonempty square prefix of $\alpha'(b^2W)$ is $\alpha'(b^2) = 11$, and the only nonempty square prefixes of $\alpha'(W)$ are $\alpha'(c^2)$ and $\alpha'(c^4)$.

To complete the proof of the theorem, we assume that

$$\beta' : (\Xi_2 \cup \{a,b,c,p,q,r,s,t\})^* \to \Sigma_2^*$$

is a nonerasing morphism such that $\beta'(V') = \alpha'(U')$ and derive a contradiction. Because $\beta'(V')$ has the nonempty square prefix $\beta'(a^2)$, and the only nonempty square prefix of $\alpha'(U')$ is 00, it must be $\beta'(a) = 0$. Similarly, we see that $\beta'(b) = 1$ and $\beta'(c) \in \{\alpha'(c), \alpha'(c^2)\}$. If $\beta'(c) = \alpha'(c^2)$, then $\beta'(c^2) = \alpha'(c^4)$ has two occurrences in $\alpha'(U')$, which is not possible, so it must be $\beta'(c) = \alpha'(c)$. It follows that

$$\beta'(sc\sigma(V)c^2pq^2rct) = \alpha'(c^2\sigma(U)c^2pq^2rc\sigma(V)c^2a^2b^2ca).$$

Here on the right-hand side, there are only two occurrences of $\alpha'(c^2)$ as an internal factor, so either

$$\beta'(sc\sigma(V)) = \alpha'(c^2\sigma(U)) \qquad \text{and} \qquad \beta'(pq^2rct) = \alpha'(pq^2rc\sigma(V)c^2a^2b^2ca) \tag{5}$$

or

$$\beta'(sc\sigma(V)) = \alpha'(c^2\sigma(U)c^2pq^2rc\sigma(V)) \qquad \text{and} \qquad \beta'(pq^2rct) = \alpha'(a^2b^2ca). \tag{6}$$

There is only one occurrence of $\alpha'(c)$ as an internal factor in $\alpha'(c^2\sigma(U))$, so (5) implies $\beta'(\sigma(V)) = \alpha'(\sigma(U)) = \alpha(U)$, which is a contradiction because $\beta' \circ \sigma$ is a nonerasing Σ_2-preserving morphism. There is only one occurrence of $\alpha'(c)$ as an internal factor in $\alpha'(a^2b^2ca)$, so (6) implies $\beta'(pq^2r) = \alpha'(a^2b^2) = 0011$, which is a contradiction because 0011 does not have a nonempty internal square factor. These contradictions show that the morphism β' does not exist, and therefore $L_{\mathrm{NE}}(U') \not\subseteq L_{\mathrm{NE}}(V')$. ◀

▶ **Theorem 11.** *For all $m, n \in \mathbb{Z}_+$,*

$$\mathrm{PatIncl}_{\mathrm{NE}}(m, n, 2) \leq_p \mathrm{PatIncl}^{\mathrm{CF}}_{\mathrm{NE}}(\max\{m, n\} + 6, n + 8, 2).$$

Proof. Let U be a pattern over (Ξ_1, Σ_2) and V a pattern over (Ξ_2, Σ_2), where $|\Xi_1| = m$ and $|\Xi_2| = n$. Because renaming the variables in one of U and V does not change the pattern language, we can assume that one of Ξ_1 and Ξ_2 is a subset of the other, and therefore $|\Xi_1 \cup \Xi_2| = \max\{m, n\}$. The constant-free patterns U' and V' defined in (1) can be constructed in polynomial time. By Lemmas 9 and 10, $L_{\mathrm{NE}}(U) \subseteq L_{\mathrm{NE}}(V)$ if and only if $L_{\mathrm{NE}}(U') \subseteq L_{\mathrm{NE}}(V')$. The claim follows. ◀

▶ **Corollary 12.** *The decision problem $\mathrm{PatIncl}^{\mathrm{CF}}_{\mathrm{NE}}(2560, 2562, 2)$ is undecidable.*

Proof. By Theorem 3, $\mathrm{PatIncl}_{\mathrm{NE}}(3, 2554, 2)$ is undecidable. It follows from Theorem 11 that $\mathrm{PatIncl}^{\mathrm{CF}}_{\mathrm{NE}}(2560, 2562, 2)$ is undecidable. ◀

4 Nonperiodic satisfiability

We are going to prove that the decision problem $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(\star, 2)$ is NP-hard. This is based on the NP-hardness of $\mathrm{EqSat}(\star, 1)$. Before the proofs, we give a brief informal explanation of the idea.

We are going to transform a system of word equations over (Ξ, Σ_1) into a similarly-behaving system of constant-free word equations over $(\Xi \cup \{a, b\}, \Sigma_2)$, where a and b are new variables. The letter 0 in the original system has two important properties: It is primitive, and the images of all variables are powers of it. We want to replace 0 by a word consisting of variables that has similar properties. We can use the word a^2b^2. For all $\{a, b\}$-nonperiodic solutions β, the word $\beta(a^2b^2)$ is primitive, and we can force $\beta(x)$ to be a power of $\beta(a^2b^2)$ by adding the equation (xa^2b^2, a^2b^2x) for all $x \in \Xi$. Finally, adding the equation (xy, yx) for all $x, y \in \Xi$ makes sure that every $\{a, b\}$-periodic solution is periodic.

▶ **Theorem 13.** *For all $n \in \mathbb{Z}_+$,*

$$\mathrm{EqSat}(n, 1) \leq_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(n + 2, 2) \qquad \textit{and} \qquad \mathrm{EqSat}(\star, 1) \leq_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(\star, 2).$$

Proof. Let S be a system of word equations over (Ξ, Σ_1). Let a, b be new variables not in Ξ. Let us define a Ξ-preserving morphism

$$\sigma : (\Xi \cup \Sigma_1)^* \to (\Xi \cup \{a, b\})^*,\ \sigma(0) = a^2b^2,$$

and a morphism

$$\tau : \{a, b\}^* \to \Sigma_2^*,\ \tau(a) = 0,\ \tau(b) = 1.$$

We can construct in polynomial time a system of constant-free word equations

$$S' = \{(\sigma(U), \sigma(V)) \mid (U, V) \in S\} \cup \{(x\sigma(0), \sigma(0)x) \mid x \in \Xi\} \cup \{(xy, yx) \mid x, y \in \Xi\}$$

over $(\Xi \cup \{a, b\}, \Sigma_2)$. To complete the proof of the theorem, we show that S' has a nonperiodic solution if and only if S has a solution.

First, assume that S has a solution α. We can define a nonperiodic Σ_2-preserving morphism

$$\beta : (\Xi \cup \{a, b\} \cup \Sigma_2)^* \to \Sigma_2^*,\ \beta(x) = \tau(x) \text{ for all } x \in \{a, b\},$$
$$\beta(x) = \tau(\sigma(\alpha(x))) \text{ for all } x \in \Xi$$

and show that it is a solution of S'. By using the definition of β and the fact that α is Σ_1-preserving and σ is Ξ-preserving, we get

$$\beta(\sigma(0)) = \tau(\sigma(0)) = \tau(\sigma(\alpha(0))),$$
$$\beta(\sigma(x)) = \beta(x) = \tau(\sigma(\alpha(x))) \text{ for all } x \in \Xi. \tag{7}$$

For all $(U, V) \in S$, from $U, V \in (\Xi \cup \Sigma_1)^*$, (7), and $\alpha(U) = \alpha(V)$, it follows that

$$\beta(\sigma(U)) = \tau(\sigma(\alpha(U))) = \tau(\sigma(\alpha(V))) = \beta(\sigma(V)).$$

Thus β is a solution of $(\sigma(U), \sigma(V))$ for all $(U, V) \in S$. For all $x \in \Xi$, we have $\alpha(x) \in 0^*$ and therefore

$$\beta(x) = \tau(\sigma(\alpha(x))) \in \tau(\sigma(0))^* = \beta(\sigma(0))^*.$$

Thus β is a solution of all the other equations in S' as well.

Second, assume that S' has a nonperiodic solution β. From $\beta(xy) = \beta(yx)$ for all $x, y \in \Xi \cup \{\sigma(0)\}$ it follows that there exists a primitive word R such that $\beta(x) \in R^*$ for all $x \in \Xi \cup \{\sigma(0)\}$. If β is $\{a, b\}$-periodic, then $\beta(\sigma(0)) = \beta(a^2b^2) \in R^*$ implies $\beta(a), \beta(b) \in R^*$, and then β is periodic, a contradiction. Therefore β must be $\{a, b\}$-nonperiodic. It follows from Theorem 7 that $\beta(\sigma(0))$ is primitive and therefore $\beta(\sigma(0)) = R$. We can define a bijective morphism

$$\phi : R^* \to \Sigma_1^*,\ \phi(R) = 0,$$

and a morphism

$$\alpha : (\Xi \cup \Sigma_1)^* \to \Sigma_1^*,\ \alpha = \phi \circ \beta \circ \sigma.$$

Then α is well-defined because the image of $\beta \circ \sigma$ is a subset of R^*, and α is Σ_1-preserving because $\alpha(0) = \phi(R) = 0$, and α is a solution of S because

$$\alpha(U) = \phi(\beta(\sigma(U))) = \phi(\beta(\sigma(V))) = \alpha(V)$$

for all $(U, V) \in S$. ◀

▶ **Corollary 14.** *The decision problem* $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(\star, 2)$ *is NP-hard.*

Proof. Follows from Theorem 13 because $\mathrm{EqSat}(\star, 1)$ is NP-hard. ◀

5 Noncommuting satisfiability

We have proved that $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}$ is NP-hard, but based on this result alone, it might be possible that, for example, $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}$ is NP-complete but EqSat is not in NP. We would like to prove that constant-free equations are, in some sense, as hard as general word equations. We are going to prove this kind of a result for the decision problem $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}$.

When trying to generalize the ideas of the previous section to the case where the alphabet of constants is Σ_k with $k > 1$, it is quite easy to define words $C_i \in \{a, b\}$ for all $i \in \Sigma_k$ so that for all $\{a, b\}$-nonperiodic morphisms β, the words $\beta(C_i)$ are distinct primitive words and $\{\beta(C_0), \ldots, \beta(C_{k-1})\}$ is the basis of a free monoid. The problem is that we cannot make sure that $\beta(x)$ is in this free monoid for all original variables x. (This difficulty is related to the fact that if $X \in \Sigma^*$, $\Gamma \subsetneq \Sigma$, $|\Gamma| \geq 2$, then the property $X \in \Gamma^*$ cannot be expressed by word equations, see [19].) However, we can make sure that $\beta(x)$ is in a certain larger free monoid whose basis contains the words $\beta(C_0), \ldots, \beta(C_{k-1})$. This is sufficient to prove the results.

▶ **Lemma 15.** *Let $k \in \mathbb{Z}_+$ and let a, b be variables. Let*

$$A_i = a^i b^2 a^{k-i+1} \qquad \text{for all } i \in \{0, \ldots, k+1\},$$
$$B = A_k^2 A_{k+1}^2 A_k^2,$$
$$C_i = B A_i B \qquad \text{for all } i \in \{0, \ldots, k-1\}.$$

Let $\beta : \{a, b\}^* \to \Sigma^*$ be an $\{a, b\}$-nonperiodic (and therefore injective) morphism and let

$$M = (\beta(B)\Sigma^* \cap \Sigma^*\beta(B)) \cup \{\varepsilon\}.$$

The following are true:

1. *$\beta(A_0), \ldots, \beta(A_{k+1}), \beta(B)$ are primitive.*
2. *M is a free monoid.*
3. *$\beta(C_0), \ldots, \beta(C_{k-1})$ are in the basis of M.*
4. *If $U \in M \setminus \{\varepsilon\}$ and $UV = VU$, then $V \in M$.*

Proof.

1. The primitivity of the words $\beta(A_i)$ follows from Theorem 7. The word $\beta(B)$ is the image of 001100 under the morphism defined by $0 \mapsto \beta(A_k)$, $1 \mapsto \beta(A_{k+1})$, and $\beta(A_k A_{k+1}) \neq \beta(A_{k+1} A_k)$ because $A_k A_{k+1} \neq A_{k+1} A_k$, so also the primitivity of $\beta(B)$ follows from Theorem 7.
2. We use Theorem 2. Clearly M is a monoid. We have to show that if $U, V, W \in \Sigma^*$ and $U, VW, UV, W \in M$, then $V \in M$. If $V = \varepsilon$, then $V \in M$. If $|V| \geq |\beta(B)|$, then VW and thus also V begins with $\beta(P)$, and UV and thus also V ends with $\beta(P)$, so $V \in M$. If $0 < |V| < |\beta(B)|$, then we can write $U = X\beta(B)$, $UV = Y\beta(B)$, $UVW = X\beta(B)^2 Z$ for some words X, Y, Z such that $|X| < |Y| < |X\beta(B)|$, so $\beta(B)$ is an internal factor of $\beta(B)^2$, which contradicts the primitivity of $\beta(P)$. Thus M is a free monoid by Theorem 2.
3. Clearly $\beta(C_i) \in M$. If $\beta(C_i)$ is not in the basis for some $i \in \{0, \ldots, k-1\}$, then it is a product of two nonempty elements of M, so it has a factor $\beta(B^2)$ and thus also a factor $\beta(A_k^4)$ and we can write

$$\beta(C_i) = \beta(A_k^2 A_{k+1}^2 A_k^2 A_i A_k^2 A_{k+1}^2 A_k^2) = U\beta(A_k^4)V. \tag{8}$$

for some words U, V. Let $l = |\beta(A_k)|$. Note that $l = |\beta(A_j)|$ for all j. If l divides $|U|$, then it follows from (8) that $\beta(A_k) = \beta(A_{k+1})$ or $\beta(A_k) = \beta(A_i)$, which contradicts the injectivity of β. If l does not divide $|U|$, then it follows from (8) that $\beta(A_k)$ is an internal factor of $\beta(A_k^2)$, which contradicts the primitivity of $\beta(A_k)$. This proves the claim.

4. If $V = \varepsilon$, then $V \in M$. If $|V| \geq |\beta(B)|$, then U and thus also V begins and ends with $\beta(B)$, so $V \in M$. If $0 < |V| < |\beta(B)|$, then $\beta(B) = R^k R'$, where R is the common primitive root of U and V, R' is a nonempty prefix of R, and $k \geq 1$. Because $\beta(B)$ is primitive, $0 < |R'| < |R|$. But R is a suffix of U and thus of $\beta(B) = R^k R'$, so R is an internal factor of R^2, which contradicts the primitivity of R. This proves the claim. ◀

▶ **Theorem 16.** *For all $n \in \mathbb{Z}_+$,*

$$\mathrm{EqSat}(n, \star) \leq_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}(2n+2, 2) \qquad \textit{and} \qquad \mathrm{EqSat}(\star, \star) \leq_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}(\star, 2).$$

Proof. Let S be a system of word equations over (Ξ, Σ_k), where $\Xi = \{x_1, \ldots, x_n\}$. Let $a, b, y_1, \ldots, y_n$ be new variables not in Ξ and let $\Xi' = \Xi \cup \{a, b, y_1, \ldots, y_n\}$. Let B, C_i, M be as in Lemma 15. Let us define a Ξ-preserving morphism

$$\sigma : (\Xi \cup \Sigma_k)^* \to (\Xi \cup \{a, b\})^*,\ \sigma(i) = C_i \text{ for all } i \in \Sigma_k,$$

and a morphism

$$\tau : \{a, b\}^* \to \Sigma_2^*,\ \tau(a) = 0,\ \tau(b) = 1.$$

We can construct in polynomial time a system of constant-free word equations

$$S' = \{(\sigma(U), \sigma(V)) \mid (U, V) \in S\} \cup \{(x_i B y_i B, B y_i B x_i) \mid i \in \{1, \ldots, n\}\}.$$

over (Ξ', Σ_2). To complete the proof of the theorem, we show that S' has an $\{a, b\}$-nonperiodic solution β if and only if S has a solution.

First, assume that S has a solution α. For all i, if $\sigma(\alpha(x_i)) = \varepsilon$, let $Y_i = \varepsilon$, and otherwise let $\sigma(\alpha(x_i)) = BY_iB$. Such words Y_i exist by the definition of σ. We can define an $\{a, b\}$-nonperiodic Σ_2-preserving morphism

$$\begin{aligned} \beta : (\Xi' \cup \Sigma_2)^* \to \Sigma_2^*,\ &\beta(x) = \tau(x) \text{ for all } x \in \{a, b\}, \\ &\beta(x_i) = \tau(\sigma(\alpha(x_i))) \text{ for all } i, \\ &\beta(y_i) = \tau(Y_i) \text{ for all } i, \end{aligned}$$

and show that it is a solution of S'. By using the definition of β and the fact that α is Σ_k-preserving and σ is Ξ-preserving, we get

$$\begin{aligned} &\beta(\sigma(i)) = \tau(\sigma(i)) = \tau(\sigma(\alpha(i))) \text{ for all } i \in \Sigma_k, \\ &\beta(\sigma(x)) = \beta(x) = \tau(\sigma(\alpha(x))) \text{ for all } x \in \Xi. \end{aligned} \tag{9}$$

For all $(U, V) \in S$, from $U, V \in (\Xi \cup \Sigma_k)^*$, (9), and $\alpha(U) = \alpha(V)$, it follows that

$$\beta(\sigma(U)) = \tau(\sigma(\alpha(U))) = \tau(\sigma(\alpha(V))) = \beta(\sigma(V)).$$

Thus β is a solution of $(\sigma(U), \sigma(V))$ for all $(U, V) \in S$. We have $\beta(x_i) = \varepsilon$ or $\beta(x_i) = \tau(BY_iB) = \beta(By_iB)$ for all i, so β is a solution of the other equations $(x_i B y_i B, B y_i B x_i)$ in S' as well.

Second, assume that S' has an $\{a, b\}$-nonperiodic solution β. By Lemma 15, from $\beta(x_i B y_i B) = \beta(B y_i B x_i)$ it follows that $\beta(x_i) \in M$ for all i. Again by Lemma 15, M is free and the words $\beta(C_i)$ are in the basis of M, so there exists an infinite alphabet Γ containing Σ_k and an injective morphism

$\phi : M \to \Gamma^*$ such that $\phi(\beta(C_i)) = i$ for all i. We can define a Σ_k-preserving morphism

$$\psi : \Gamma^* \to \Sigma_k^*,\ \psi(x) = \varepsilon \text{ for all } x \in \Gamma \setminus \Sigma_k$$

and a morphism

$$\alpha : (\Xi \cup \Sigma_k)^* \to \Sigma_k^*,\ \alpha = \psi \circ \phi \circ \beta \circ \sigma.$$

Then α is well-defined because $\beta(\sigma(x)) \in M$ for all $x \in \Xi \cup \Sigma_k$, and α is Σ_k-preserving because

$$\alpha(i) = \psi(\phi(\beta(C_i))) = \psi(i) = i$$

for all $i \in \Sigma_k$, and α is a solution of S because from $\beta(\sigma(U)) = \beta(\sigma(V))$ it follows that

$$\alpha(U) = \psi(\phi(\beta(\sigma(U)))) = \psi(\phi(\beta(\sigma(V)))) = \alpha(V)$$

for all $(U, V) \in S$. ◀

▶ **Corollary 17.** $\mathrm{EqSat}(\star, \star) \equiv_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}(\star, \star)$.

Proof. We proved in Theorem 16 that $\mathrm{EqSat}(\star, \star) \leq_p \mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NC}}(\star, \star)$. To prove the other direction, let S be a system of constant-free equations over (Ξ, Σ_k), $k \geq 2$, and $x, y \in \Xi$. Let p, q, r be new variables not in Ξ. It is easy to see that the system

$$S' = S \cup \{(xy, p0q), (yx, p1r)\}$$

over $(\Xi \cup \{p, q, r\}, \Sigma_k)$ has a solution if and only if S has a $\{x, y\}$-nonperiodic solution: If β is a solution of S', then the restriction of β on $(\Xi \cup \Sigma_k)^*$ is a solution of S, and it is $\{x, y\}$-nonperiodic because

$$\beta(xy) = \beta(p)0\beta(q) \neq \beta(p)1\beta(r) = \beta(yx).$$

On the other hand, if α is an $\{x, y\}$-nonperiodic solution of S, then we can write $\alpha(xy) = PaQ$ and $\alpha(yx) = PbR$ for some words P, Q, R and distinct letters a, b. Because S is constant-free, every morphism we get from α by permuting the constant letters in the images of the variables is also an $\{x, y\}$-nonperiodic solution of S, so we can assume that $a = 0$ and $b = 1$. Then we can extend α to a solution α' of S' by defining $\alpha'(p) = P$, $\alpha'(q) = Q$, $\alpha'(r) = R$. This completes the proof. ◀

6 Conclusion

We have proved that the inclusion problem of nonerasing pattern languages is undecidable even in the case of constant-free patterns. We have also proved that the nonperiodic satisfiability problem of constant-free word equations is NP-hard, and the noncommuting satisfiability problem of constant-free word equations is polynomially equivalent to the general satisfiability problem of word equations.

The following questions remain open:

- Is the equivalence problem of erasing pattern languages decidable?
- Is the satisfiability problem of word equations in NP?
- For some fixed $n \geq 3$, can we prove that $\mathrm{EqSat}(n, \star)$ is in P or NP or NP-hard?

There are also several smaller open questions raised by the new results:

- Can the numbers 2560 and 2562 in Corollary 12 be made significantly smaller? This would require either a rather different approach or improving the results in [4].
- Is $\mathrm{EqSat}(\star,\star)$ polynomial-time reducible to $\mathrm{EqSat}^{\mathrm{CF}}_{\mathrm{NP}}(\star,\star)$?
- In Theorem 16, we used $n+2$ new variables. Would a constant number of new variables be sufficient?
- The satisfiability problem remains NP-hard for several restricted subfamilies of word equations. Can we prove NP-hardness results for some interesting subfamilies of constant-free equations?

References

1 Dana Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21(1):46–62, 1980. doi:10.1016/0022-0000(80)90041-0.

2 Evelyne Barbin-Le Rest and Michel Le Rest. Sur la combinatoire des codes à deux mots. *Theoretical Computer Science*, 41(1):61–80, 1985. doi:10.1016/0304-3975(85)90060-X.

3 Jean Berstel, Dominique Perrin, and Christophe Reutenauer. *Codes and Automata*. Cambridge University Press, 2010.

4 Joachim Bremer and Dominik D. Freydenberger. Inclusion problems for patterns with a bounded number of variables. *Information and Computation*, 220/221:15–43, 2012. doi:10.1016/j.ic.2012.10.003.

5 Christian Choffrut and Juhani Karhumäki. Combinatorics of words. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 329–438. Springer, 1997. doi:10.1007/978-3-642-59136-5_6.

6 Joel D. Day, Florin Manea, and Dirk Nowotka. The hardness of solving simple word equations. In *Proceedings of the 42nd MFCS*, volume 83 of *LIPIcs*, pages 18:1–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.MFCS.2017.18.

7 Robert Dąbrowski and Wojtek Plandowski. Solving two-variable word equations (extended abstract). In *Proceedings of the 31st ICALP*, volume 3142 of *LNCS*, pages 408–419. Springer, 2004. doi:10.1007/978-3-540-27836-8_36.

8 Volker Diekert. Makanin's algorithm. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, pages 387–442. Cambridge University Press, 2002.

9 Volker Diekert and Murray Elder. Solutions of twisted word equations, EDT0L languages, and context-free groups. In *Proceedings of the 44th ICALP*, volume 80 of *LIPIcs*, pages 96:1–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.96.

10 Henning Fernau and Markus L. Schmid. Pattern matching with variables: a multivariate complexity analysis. *Information and Computation*, 242:287–305, 2015. doi:10.1016/j.ic.2015.03.006.

11 Dominik D. Freydenberger and Daniel Reidenbach. Bad news on decision problems for patterns. *Information and Computation*, 208(1):83–96, 2010. doi:10.1016/j.ic.2009.04.002.

12 Tero Harju and Juhani Karhumäki. The equivalence problem of multitape finite automata. *Theoretical Computer Science*, 78(2):347–355, 1991. doi:10.1016/0304-3975(91)90356-7.

13 Tero Harju, Juhani Karhumäki, and Wojciech Plandowski. Independent systems of equations. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, pages 443–472. Cambridge University Press, 2002.

14 Ju. I. Hmelevskiĭ. *Equations in free semigroups*. American Mathematical Society, 1976. Translated by G. A. Kandall from the Russian original: Trudy Mat. Inst. Steklov. 107 (1971).

15 Artur Jeż. One-variable word equations in linear time. *Algorithmica*, 74(1):1–48, 2016. doi:10.1007/s00453-014-9931-3.

16 Artur Jeż. Recompression: a simple and powerful technique for word equations. *Journal of the ACM*, 63(1):Art. 4, 51, 2016. doi:10.1145/2743014.

17 Artur Jeż. Word equations in nondeterministic linear space. In *Proceedings of the 44th ICALP*, volume 80 of *LIPIcs*, pages 95:1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.95.

18 Tao Jiang, Arto Salomaa, Kai Salomaa, and Sheng Yu. Decision problems for patterns. *Journal of Computer and System Sciences*, 50(1):53–63, 1995. doi:10.1006/jcss.1995.1006.

19 Juhani Karhumäki, Filippo Mignosi, and Wojciech Plandowski. The expressibility of languages and relations by word equations. *Journal of the ACM*, 47(3):483–505, 2000. doi:10.1145/337244.337255.

20 Juhani Karhumäki and Wojciech Plandowski. On the defect effect of many identities in free semigroups. In Gheorghe Paun, editor, *Mathematical aspects of natural and formal languages*, pages 225–232. World Scientific, 1994. doi:10.1142/9789814447133_0012.

21 M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002. URL: http://www-igm.univ-mlv.fr/~berstel/Lothaire/AlgCWContents.html.

22 Roger C. Lyndon and Marcel-Paul Schützenberger. The equation $a^M = b^N c^P$ in a free group. *The Michigan Mathematical Journal*, 9(4):289–298, 1962. doi:10.1307/mmj/1028998766.

23 G. S. Makanin. The problem of the solvability of equations in a free semigroup. *Mat. Sb. (N.S.)*, 103(2):147–236, 1977. English translation in Math. USSR Sb. 32:129–198, 1977.

24 Florin Manea and Markus L. Schmid. Matching patterns with variables. In *Proceedings of the 12th WORDS*, volume 11682 of *LNCS*, pages 1–27. Springer, 2019. doi:10.1007/978-3-030-28796-2_1.

25 Yen Kaow Ng and Takeshi Shinohara. Developments from enquiries into the learnability of the pattern languages from positive data. *Theoretical Computer Science*, 397(1–3):150–165, 2008. doi:10.1016/j.tcs.2008.02.028.

26 Dirk Nowotka and Aleksi Saarela. An optimal bound on the solution sets of one-variable word equations and its consequences. In *Proceedings of the 45th ICALP*, volume 107 of *LIPIcs*, pages 136:1–136:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.136.

27 Wojciech Plandowski. Satisfiability of word equations with constants is in PSPACE. *Journal of the ACM*, 51(3):483–496, 2004.

28 Daniel Reidenbach. Discontinuities in pattern inference. *Theoretical Computer Science*, 397(1–3):166–193, 2008. doi:10.1016/j.tcs.2008.02.029.

29 John Michael Robson and Volker Diekert. On quadratic word equations. In *Proceedings of the 16th STACS*, volume 1563 of *LNCS*, pages 217–226. Springer, 1999. doi:10.1007/3-540-49116-3_20.

30 Aleksi Saarela. On the complexity of Hmelevskii's theorem and satisfiability of three unknown equations. In *Proceedings of the 13th DLT*, volume 5583 of *LNCS*, pages 443–453. Springer, 2009. doi:10.1007/978-3-642-02737-6_36.

31 Aleksi Saarela. Studying word equations by a method of weighted frequencies. *Fundamenta Informaticae*, 162(2–3):223–235, 2018. doi:10.3233/FI-2018-1722.

32 Géraud Sénizergues. The equivalence problem for deterministic pushdown automata is decidable. In *Proceedings of the 24th ICALP*, volume 1256 of *LNCS*, pages 671–681. Springer, 1997. doi:10.1007/3-540-63165-8_221.

33 Takeshi Shinohara. Polynomial time inference of extended regular pattern languages. In *RIMS Symposia on Software Science and Engineering*, volume 147 of *LNCS*, pages 115–127. Springer, 1983. doi:doi.org/10.1007/3-540-11980-9_19.

34 Jean-Claude Spehner. *Quelques problémes d'extension, de conjugaison et de présentation des sous-monoïdes d'un monoïde libre*. PhD thesis, Univ. Paris, 1976.