The Hardness of Solving Simple Word Equations

Joel D. Day[1], Florin Manea[2], and Dirk Nowotka[3]

1 **Kiel University, Department of Computer Science, Kiel, Germany**
jda@informatik.uni-kiel.de
2 **Kiel University, Department of Computer Science, Kiel, Germany**
flm@informatik.uni-kiel.de
3 **Kiel University, Department of Computer Science, Kiel, Germany**
dn@informatik.uni-kiel.de

Abstract

We investigate the class of regular-ordered word equations. In such equations, each variable occurs at most once in each side and the order of the variables occurring in both left and right hand sides is preserved (the variables can be, however, separated by potentially distinct constant factors). Surprisingly, we obtain that solving such simple equations, even when the sides contain exactly the same variables, is NP-hard. By considerations regarding the combinatorial structure of the minimal solutions of the more general quadratic equations we obtain that the satisfiability problem for regular-ordered equations is in NP. The complexity of solving such word equations under regular constraints is also settled. Finally, we show that a related class of simple word equations, that generalises one-variable equations, is in P.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, F.4.3 Formal Languages

Keywords and phrases Word Equations, Regular Patterns, Regular Constraints

Digital Object Identifier 10.4230/LIPIcs.MFCS.2017.18

1 Introduction

A *word equation* is an equality $\alpha = \beta$, where α and β are words over an alphabet $\Sigma \cup X$ (called the left, respectively, right side of the equation); $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ is the alphabet of *constants* and $X = \{x_1, x_2, x_3, \ldots\}$ is the alphabet set of *variables*. A *solution* to the equation $\alpha = \beta$ is a morphism $h : (\Sigma \cup X)^* \to \Sigma^*$ that acts as the identity on Σ and satisfies $h(\alpha) = h(\beta)$. For instance, $\alpha = x_1\mathtt{ab}x_2$ and $\beta = \mathtt{a}x_1x_2\mathtt{b}$ 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 $\ell \geq 0$.

The study of word equations (or the existential theory of equations over free monoids) is an important topic found at the intersection of algebra and computer science, with significant connections to, e.g., combinatorial group or monoid theory [19, 18, 2], unification [25, 11, 12], and, more recently, data base theory [9, 8]. The problem of deciding whether a given word equation $\alpha = \beta$ has a solution or not, known as the satisfiability problem, was shown to be decidable by Makanin [20] (see Chapter 12 of [17] for a survey). Later it was shown that the satisfiability problem is in PSPACE by Plandowski [22]; a new proof of this result was obtained in [14], based on a new simple technique called recompression. However, it is conjectured that the satisfiability problem is in NP; this would match the known lower bounds: the satisfiability of word equations is NP-hard, as it follows immediately from, e.g., [4]. This hardness result holds in fact for much simpler classes of word equations, like quadratic equations (where the number of occurrences of each variable in $\alpha\beta$ is at most two), as shown in [3]. There are also cases when the satisfiability problem is tractable. For

42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017).
Editors: Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin; Article No. 18; pp. 18:1–18:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

instance, word equations with only one variable can be solved in linear time in the size of the equation, see [13]; equations with two variables can be solved in time $\mathcal{O}(|\alpha\beta|^5)$, see [1].

In most cases, the NP-hardness of the satisfiability problem for classes of word equations was shown as following from the NP-completeness of the *matching problem* for corresponding classes of patterns with variables. In the matching problem we essentially have to decide whether an equation $\alpha = \beta$, with $\alpha \in (\Sigma \cup X)^*$ and $\beta \in \Sigma^*$, has a solution; that is, only one side of the equation, called pattern, contains variables. The aforementioned results [4, 3] show, in fact, that the matching problem is NP-complete for general α, respectively when α is quadratic. Many more tractability and intractability results concerning the matching problem are known (see [24, 6, 7]). In [5], efficient algorithms were defined for, among others, patterns which are *regular* (each variable has at most one occurrence), *non-cross* (between any two occurrences of a variable, no other distinct variable occurs), or patterns with only a constant number of variables occurring more than once.

Naturally, for a class of patterns that can be matched efficiently, the hardness of the satisfiability problem for word equations with sides in the respective class is no longer immediate. A study of such word equations was initiated in [21], where the following results were obtained. Firstly, the satisfiability problem for word equations with non-cross sides (for short non-cross equations) remains NP-hard. In particular, solving non-cross equations $\alpha = \beta$ where each variable occurs at most three times, at most twice in α and exactly once in β, is NP-hard. Secondly, the satisfiability of one-repeated variable equations (where at most one variable occurs more than once in $\alpha\beta$, but arbitrarily many other variables occur only once) having at least one non-repeated variable on each side, was shown to be trivially in P.

In this paper we mainly address the class of regular-ordered equations, whose sides are regular patterns and, moreover, the order of the variables occurring in both sides is the same. This seems to be one of the structurally simplest classes of equations whose number of variables is not bounded by a constant. One central motivation for studying these equations with a simple structure is that understanding their complexity and combinatorial properties may help us to identify a boundary between classes of word equations whose satisfiability is tractable and intractable. Moreover, we wish to gain a better understanding of the core reasons why solving word equations is hard. In the following, we overview our results, methods, and their connection to existing works from the literature.

Lower bounds. Our first result closes the main problem left open in [21]. Namely, we show that it is (still) NP-hard to solve regular (ordered) word equations. Note that in these word equations each variable occurs at most twice: at most once in every side. They are particular cases of both quadratic equations and non-cross equations, so the reductions showing the hardness of solving these more general equations do not carry over. To begin with, matching quadratic patterns is NP-hard, while matching regular patterns can be done in linear time. Showing the hardness of the matching problem for quadratic patterns in [3] relied on a simple reduction from 3-SAT, where the two occurrences of each variable were used to simulate an assignment of a corresponding variable in the SAT formula, respectively to ensure that this assignment satisfies the formula. To facilitate this final part, the second occurrences of the variables were grouped together, so the equation constructed in this reduction was not non-cross. Indeed, matching non-cross patterns can be done in polynomial time. So showing that solving non-cross equations is hard, in [21], required slightly different techniques. This time, the reduction was from an assignment problem in graphs. The (single) occurrences of the variables in one side of the equation were used to simulate an assignment in the graph, while the (two) occurrences of the variables from the other side were used for two reasons:

to ensure that the previously mentioned assignment is correctly constructed and to ensure that it also satisfies the requirements of the problem. For the second part it was also useful to allow the variables to occur in one side in a different order as in the other side.

As stated in [21], showing that the satisfiability problem for regular equations seems to require a totally different approach. Our hardness reduction relies on some novel ideas, and, unlike the aforementioned proofs, has a deep word-combinatorics core. As a first step, we define a reachability problem for a certain type of (regulated) string rewriting systems, and show it is NP-complete (in Lemma 7). This is achieved via a reduction from the strongly NP-complete problem 3-Partition [10]. Then we show that this reachability problem can be reduced to the satisfiability of regular-ordered word equations; in this reduction (described in the successive Lemmas 9, 10, and 11), we essentially try to encode the applications of the rewriting rules of the system into the periods of the words assigned to the variables in a solution to the equation. In doing this, we are able to only use one occurrence of each variable per side, and moreover to even have the variables in the same order in both sides.

Our reduction suggests the ability of this simple class of equations to model other natural problems in rewriting, combinatorics on words, and beyond. In this respect, our construction seems interesting with respect to the expressibility of word equations, as studied in [15].

Upper bounds. A consequence of the results in [23] is that the satisfiability problem for a certain class of word equations is in NP if the lengths of the minimal solutions of such equations (where the length of the solution defined by a morphism h is the image of the equation's sides under h) are at most exponential. With this in mind, we show Lemma 14, which gives us an insight in the combinatorial structure of the minimal solutions of quadratic equations: if we follow around the minimal solutions the positions that are fixed inside the images of the variables by each terminal of the original equation (in order, starting with that terminal), we obtain sequences that should not contain repetitions. Consequently, in Proposition 17, we give a simple and concise proof of the fact that the image of any variable in a minimal solution to a regular-ordered equation is at most linear in the size of the equation. It immediately follows that the satisfiability problem for regular-ordered equations is in NP. While this result was expected, the approach we use to obtain it seems rather interesting to us, and also a promising approach to showing that other, more complicated, classes of restricted word equations can be solved in NP-time. For instance, it is an open problem to show this for arbitrary regular or quadratic equations. It is worth noting that our polynomial upper bound on length of minimal solutions of regular-ordered equations does not hold even for slightly relaxed versions of such equations. More precisely, non-cross equations $\alpha = \beta$ where the order of the variables is the same in both sides and each variable occurs exactly three times in $\alpha\beta$, but never only on one side, may already have exponentially long minimal solutions (see Proposition 2). To this end, it seems even more surprising that it is NP-hard to solve equations with such a simple structure (regular-ordered), which, moreover, have quadratically short solutions. As such, regular-ordered equations seem to be among the structurally simplest word-equations, whose satisfiability problem is intractable.

Extending our ideas, we settle the complexity of solving regular-ordered equations with regular constraints (as defined in [3], where each variable is associated with an NFA), which is in NP for regular-ordered equations whose sides contain exactly the same variables, or when the languages defining the scope of the variables are all accepted by NFAs with at most c states, where c is a constant. For regular-ordered equations with regular constraints without these restrictions, the problem remains PSPACE-complete.

Finally, we use again a reasoning on the structure of the minimal solutions of equations, similar to the above, to show that if we preserve the non-cross structure of the sides of the considered word equations, but allow only one variable to occur an arbitrary number of times (all the others occur exactly once in total, and hence in at most one side), we get a class of equations whose satisfiability problem is in P. This problem is related to the one-repeated variable equations considered in [21]; in this case, we restrict the equations to a non-cross structure of the sides, but drop the condition that at least one non-repeated variable should occur on each side. Moreover, this problem generalises the one-variable equations [13], while preserving the tractability of their satisfiability problem. Last, but not least, this result shows that the pattern searching problem, in which, given a pattern $\alpha \in (\Sigma \cup \{x_1\})^*$ containing constants and exactly one variable x_1 (occurring several times) and a text $\beta \in (\Sigma \cup \{x_1\})^*$ containing constants and the same single (repeated) variable, we check whether there exists an assignment of x_1 that makes α a factor of β, is tractable; indeed, this problem is the same as checking whether the word equation $x_2 \alpha x_3 = \beta$, with $\alpha, \beta \in (\Sigma \cup \{x_1\})^*$, is satisfiable.

2 Preliminaries

Let Σ be an *alphabet*. We denote by Σ^* the set of all words over Σ; by ε we denote the empty word. Let $|w|$ denote the *length* of a word w. For $1 \leq i \leq j \leq |w|$ we denote by $w[i]$ the letter on the i^{th} position of w and $w[i..j] = w[i]w[i+1] \cdots w[j]$. A word w is *p-periodic* for $p \in \mathbb{N}$ (and p is called a period of w) if $w[i] = w[i+p]$ for all $1 \leq i \leq |w| - p$; the smallest period of a word is called its *period*. By extension, for a word w of period p, we sometimes call $w[1..p]$ the period of w. Let $w = xyz$ for some words $x, y, z \in \Sigma^*$, then x is called *prefix* of w, y is a *factor* of w, and z is a *suffix* of w. Two words w and u are called *conjugate* if there exist non-empty words x, y such that $w = xy$ and $u = yx$. The powers of w are defined by $w^0 = \varepsilon$, $w^n = ww^{n-1}$ for $n \geq 1$, and $w^\omega = ww\cdots$, an infinite concatenation of the word w.

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be an alphabet of *constants* and let $X = \{x_1, x_2, x_3, \ldots\}$ be an alphabet of *variables*. We assume that X and Σ are disjoint. A word $\alpha \in (\Sigma \cup X)^*$ is usually called *pattern*. For a pattern α and a letter $z \in \Sigma \cup X$, let $|\alpha|_z$ denote the number of occurrences of z in α; $\mathsf{var}(\alpha)$ denotes the set of variables from X occurring in α. A morphism $h : (\Sigma \cup X)^* \to \Sigma^*$ with $h(a) = a$ for every $a \in \Sigma$ is called a *substitution*. We say that $\alpha \in (\Sigma \cup X)^*$ is *regular* if, for every $x \in \mathsf{var}(\alpha)$, 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, hence the name of such patterns. The pattern α is *non-cross* if between any two occurrences of the same variable x no other variable different from x occurs, e.g., $\mathtt{a}x_1\mathtt{ba}x_1x_2\mathtt{a}x_2x_2\mathtt{b}$ is non-cross, but $x_1\mathtt{b}x_2x_2\mathtt{b}x_1$ is not.

A *word equation* is a tuple $(\alpha, \beta) \in (\Sigma \cup X)^+ \times (\Sigma \cup X)^+$; we usually denote such an equation by $\alpha = \beta$, where α is the left-hand side (LHS, for short) and β the right-hand side (RHS) of the equation. A *solution* to an equation $\alpha = \beta$ is a substitution h with $h(\alpha) = h(\beta)$, and $h(\alpha)$ is called the *solution word* (*defined by* h); the length of a solution h of the equation $\alpha = \beta$ is $|h(\alpha)|$. A solution of shortest length to an equation is also called *minimal*.

A word equation is *satisfiable* if it has a solution and the *satisfiability problem* is to decide for a given word equation whether or not it is satisfiable. The satisfiability problem for general word equations can be solved non-deterministically in time polynomial in $n \log N$, where n is the length of the equation and N the length of its minimal solution [23]. The next result follows.

▶ **Lemma 1.** *Let $\mathcal{E}$ be a class of word equations. Suppose there exists a polynomial P such that for any equation in $\mathcal{E}$ its minimal solution, if it exists, has length at most $2^{P(n)}$ where n is the length of the equation. Then the satisfiability problem for $\mathcal{E}$ is in* NP.

A word equation $\alpha = \beta$ is *regular* or *non-cross*, if both α and β are regular or both α and β are non-cross, respectively; $\alpha = \beta$ is *quadratic* if each variable occurs at most twice in $\alpha\beta$. We call a regular or non-cross 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 α and β, then all occurrences of x occur before all occurrences of y in α if and only if all occurrences of x occur before all occurrences of y in β. Note, however, that variables may occur only in one side of a regular or non-cross ordered equation. For instance $x_1x_1\mathtt{a}x_2x_3\mathtt{b}x_4 = x_1\mathtt{a}x_1x_2\mathtt{b}x_3$ is ordered non-cross, while $x_1x_1\mathtt{a}x_3x_2\mathtt{b} = x_1\mathtt{a}x_1x_2\mathtt{b}x_3$ is still non-cross but not ordered. Next we give an example of very simple word equations whose minimal solution has exponential length, whose structure follows that in [16, Theorem 4.8].

▶ **Proposition 2.** *The minimal solution to the word equation* $x_n\mathtt{a}x_n\mathtt{b}x_{n-1}\mathtt{b}x_{n-2}\cdots\mathtt{b}x_1 = \mathtt{a}x_nx_{n-1}^2\mathtt{b}x_{n-2}^2\mathtt{b}\ldots\mathtt{b}x_1^2\mathtt{b}\mathtt{a}^2$ *has length* $\Theta(2^n)$.

For a word equation $\alpha = \beta$ and an $x \in \mathsf{var}(\alpha\beta)$, a *regular constraint* (*for* x) is a regular language L_x. A solution h for $\alpha = \beta$ *satisfies* the regular constraint L_x if $h(x) \in L_x$. The satisfiability problem for word equations with regular constraints is to decide on whether an equation $\alpha = \beta$ with regular constraints L_x, $x \in \mathsf{var}(\alpha\beta)$, given as an NFA, has a solution that satisfies all regular constraints.

Finally, we recall the 3-Partition problem (see [10]). This problem is NP-complete in the strong sense, i.e., it remains NP-hard even when the input numbers are given in unary.

▶ **Problem 3** (3-Partition – 3-PAR).
Instance: $3m$ nonnegative integers (given in unary) $A = (k_1, \ldots, k_{3m})$, whose sum is ms
Question: Is there a partition of A into m disjoint groups of three elements, such that each group sums exactly to s.

3 Lower bounds

In this section, we show that the highly restricted class of regular-ordered word equations is NP-hard, and, thus, that even when the order in which the variables occur in an equation is fixed, and each variable may only repeat once – and never on the same side of the equation – satisfiability remains intractable. As mentioned in the introduction, our result shows the intractability of the satisfiability problem for a class of equations considerably simpler than the simplest intractable classes of equations known so far. Our result seems also particularly interesting since we are able to provide a corresponding upper bound in the next section, and even show that the minimal solutions of regular-ordered equations are "optimally short".

▶ **Theorem 4.** *The satisfiability problem for regular-ordered word equations is* NP*-hard.*

In order to show NP-hardness, we shall provide a reduction from a reachability problem for a simple type of regulated string-rewriting system. Essentially, given two words – a starting point, and a target – and an ordered series of n rewriting rules (a rewriting program, in a sense), the problem asks whether this series of rules may be applied consecutively (in the predefined order) to the starting word such that the result matches the target. We stress that the order of the rules is predefined, but the place where a rule is to be applied within the sentential form is non-deterministically chosen.

▶ **Problem 5** (Rewriting with Programmed Rules – REP).
Instance: Words $u_{start}, u_{end} \in \Sigma^*$ and an ordered series of n substitution rules $w_i \to w_i'$, with $w_i, w_i' \in \Sigma^*$, for $1 \le i \le n$.

Question: Can u_{end} be obtained from u_{start} by applying each rule (i.e., replacing an occurrence of w_i with w'_i), in order, to u_{start}.

▶ **Example 6.** Let $u_{start} = b^5$ and $u_{end} = (a^{11}bc^2)^5$; for $1 \leq i \leq 10$, consider the rules $w_i \to w'_i$ with $w_i = b$ and $w'_i = a^i bc$. We can obtain u_{end} from u_{start} by first applying $w_1 \to w'_1$ to the first b, then $w_2 \to w'_2$ to the second b, and further, in order for $3 \leq i \leq 5$, by applying $w_i \to w'_i$ to the i^{th} b. Then, we apply w_6 to the fifth b (counting from left to right). Further we apply in order, for $7 \leq i \leq 10$, $w_i \to w'_i$ to the $(11-i)^{th}$ occurrence of b.

It is not so hard to see that REP is NP-complete (the size of the input is the sum of the lengths of u_{start}, u_{end}, w_i and w'_i). A reduction can be given from 3-PAR, in a manner similar to the construction in the example above. Importantly for our proof, 3-PAR is strongly NP-complete, so it is simpler to reduce it to a problem whose input consists of words.

▶ **Lemma 7.** REP *is* NP*-complete.*

Our reduction centres on the construction, for any instance μ of REP, of a regular-ordered word equation $\alpha_\mu = \beta_\mu$ which possesses a specific form of solution – which we shall call *overlapping* – if and only if the instance of REP has a solution. By restricting the form of solutions in this way, the exposition of the rest of the reduction is simplified considerably. The main idea is that the applications of the rewriting rules are encoded in the periods of each variable in the solution.

▶ **Definition 8.** Let $n \in \mathbb{N}$, μ be an instance of REP with u_{start}, u_{end} and rules $w_i \to w'_i$ for $1 \leq i \leq n$. Let # be a 'new' letter not occurring in any component of REP. We define the regular-ordered equation $\alpha_\mu = \beta_\mu$ such that:
$\alpha_\mu := x_1\ w_1\ x_2\ w_2\ \cdots\ x_n\ w_n\ x_{n+1}\ \#\ u_{end}$, $\beta_\mu := \#\ u_{start}\ x_1\ w'_1\ x_2\ w'_2\ \cdots\ \ x_n\ w'_n\ x_{n+1}$.
A solution $h : (X \cup \Sigma)^* \to \Sigma^*$ is called *overlapping* if, for every $1 \leq i \leq n$, there exists $z_i \in \Sigma^*$ such that $w_i z_i$ is a suffix of $h(x_i)$ and $h(\# u_{start} x_1 \cdots w'_{i-1} x_i) = h(x_1 w_1 \cdots x_i w_i) z_i$.

Of course, satisfiability of a class of word equations asks whether any solution exists, rather than just overlapping solutions. Hence, before we prove our claim that $\alpha_\mu = \beta_\mu$ has an overlapping solution if and only if μ satisfies REP, we present a construction of an equation $\alpha = \beta$ which has a solution if and only if $\alpha_\mu = \beta_\mu$ has an overlapping solution. Essentially, this shows that solving the satisfiability of regular-ordered equations is as hard as solving the satisfiability of word equations when we restrict our search to overlapping solutions.

▶ **Lemma 9.** *Let μ be an instance of* REP*. There exists a regular-ordered equation $\alpha = \beta$ of size $O(|\alpha_\mu \beta_\mu|)$ such that $\alpha = \beta$ is satisfiable if and only if $\alpha_\mu = \beta_\mu$ has an overlapping solution.*

The rest of the proof relies on the following technical characterisation of overlapping solutions to $\alpha_\mu = \beta_\mu$ in terms of periods v_i of the images $h(x_i)$. The y_i factors will correspond to the z_i factors discussed in the definition of overlapping solutions (see also Figure 1).

▶ **Lemma 10.** *Let μ be a an instance of* REP *with u_{start}, u_{end} and rules $w_i \to w'_i$ for $1 \leq i \leq n$. A substitution $h : (X \cup \Sigma)^* \to \Sigma^*$ is an overlapping solution to $\alpha_\mu = \beta_\mu$ if and only if there exist prefixes $v_1, v_2, \ldots, v_n$ of $h(x_1), h(x_2), \ldots, h(x_n)$ such that:*
1. *$h(x_i)\ w_i$ is a prefix of v_i^ω for $1 \leq i \leq n$, and*
2. *$v_1 = \# u_{start}$, and for $2 \leq i \leq n$, $v_i = y_{i-1} w'_{i-1}$, and*
3. *$y_n\ w'_n\ h(x_{n+1}) = h(x_{n+1})\ \# u_{end}$,*

where for $1 \leq i \leq n$, y_i is the suffix of $h(x_i)$ of length $|v_i| - |w_i|$.

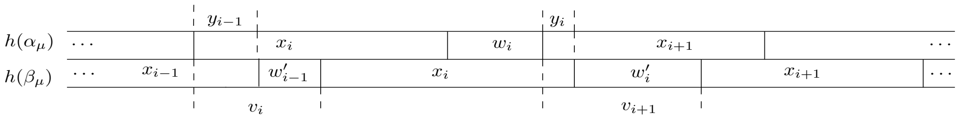

Figure 1 The period of $h(x_i)$ is v_i, and since w_iy_i is a suffix of $h(x_i)$ and $|w_iy_i| = |v_i|$, we have that w_iy_i is a cyclic shift of v_i (i.e., they are conjugate) – so $v_i = sw_it$ and $w_iy_i = w_its$ for some s, t. v_{i+1} is conjugate to sw'_it since $v_{i+1} = y_iw'_i = tsw'_i$. Thus v_{i+1} is obtained from v_i by "applying" the rule $w_i \to w'_i$ (and conjugating, but we can keep track of this due to the unique occurrence of #).

We shall now take advantage of Lemma 10 in order to demonstrate the correctness of our construction of $\alpha_\mu = \beta_\mu$ – i.e., that it has an overlapping solution if and only if μ satisfies REP. In particular, the periods v_i of the images $h(x_i)$ of the variables – which are obtained as the 'overlap' between the two occurrences of $h(x_i)$ – store the i^{th} stage of a rewriting $u_{start} \to \ldots \to u_{end}$. In fact, this is obtained as the conjugate of v_i starting with #. Thus the solution-word, when it exists, stores a sort-of rolling computation history.

▶ **Lemma 11.** *Let μ be a an instance of* REP *with u_{start}, u_{end} and rules $w_i \to w'_i$ for $1 \leq i \leq n$. There exists an overlapping solution $h : (X \cup \Sigma)^* \to \Sigma^*$ to the equation $\alpha_\mu = \beta_\mu$ if and only if μ satisfies* REP.

Proof. Suppose firstly that μ satisfies REP. Then there exist $s_1, s_2, \ldots, s_n, t_1, t_2, \ldots, t_n$ such that $u_{start} = s_1w_1t_1$, for $1 \leq i \leq n-1$, $s_iw'_it_i = s_{i+1}w_{i+1}t_{i+1}$ and $s_nw'_nt_n = u_{end}$. Let $h : (X \cup \Sigma)^* \to \Sigma^*$ be the substitution such that $h(x_1) = \#s_1w_1t_1\#s_1$, $h(x_{n+1}) = t_n\#s_nw'_nt_n\#s_nw'_nt_n$, and for $2 \leq i \leq n$, $h(x_i) = t_{i-1}\#s_{i-1}w'_{i-1}t_{i-1}\#s_i$. We shall now show that h satisfies Lemma 10, and hence that h is an overlapping solution to $\alpha_\mu = \beta_\mu$.

Let $v_1 = \#s_1w_1t_1$, let $y_1 := t_1\#s_1$, and for $2 \leq i \leq n$, let $v_i := t_{i-1}\#s_{i-1}w'_{i-1}$ and let $y_i := t_i\#s_i$. Note that for $1 \leq i \leq n$, v_i is a prefix of $h(x_i)$, and moreover, since $s_{i-1}w'_{i-1}t_{i-1} = s_iw_it_i$, y_i is the suffix of $h(x_i)$ of length $|v_i| - |w_i|$.

It is clear that h satisfies Condition (1) of Lemma 10 for $i = 1$. For $2 \leq i \leq n$, we have $h(x_i)w_it_i = t_{i-1}\#s_{i-1}w_{i-1}t_{i-1}\#s_iw_it_i = t_{i-1}\#s_{i-1}w_{i-1}t_{i-1}\#s_{i-1}w_{i-1}t_{i-1}$, which is a prefix of v_i^ω, and hence $h(x_i)w_i$ is also a prefix of v_i^ω. Thus h satisfies Condition (1) for all i. Moreover, $v_1 = \#u_{start}$, and for $2 \leq i \leq n$, $y_{i-1}w'_{i-1} = t_{i-1}\#s_{i-1}w'_{i-1} = v_i$, so h satisfies Condition (2). Finally, $y_nw'_nh(x_{n+1}) = t_n\#s_nw'_nt_n\#s_nw'_nt_n\#s_nw'_nt_n = h(x_{n+1})\#u_{end}$ so h also satisfies Condition (3).

Now suppose that h is an overlapping solution to $\alpha_\mu = \beta_\mu$. Then h satisfies Conditions (1), (2) and (3) of Lemma 10. Let v_i, y_i be defined according to the lemma for $1 \leq i \leq n$, and let $v_{n+1} = y_nw'_n$. We shall show that μ satisfies REP as follows. We begin with the following observations, whose proofs are omitted due to space constraints.

▶ **Claim 12.** *For $1 \leq i \leq n$, y_iw_i and v_i are conjugate. Hence, for $1 \leq i \leq n+1$, $|v_i|_\# = 1$.*

Let $\tilde{v}_i$ be the (unique) conjugate of v_i which has # as a prefix. Then we have the following:

▶ **Claim 13.** *For $1 \leq i \leq n$, there exist s_i, t_i such that $\tilde{v}_i = \#s_iw_it_i$ and $\tilde{v}_{i+1} = \#s_iw'_it_i$.*

Recall from Condition (3) of Lemma 10 that $v_{n+1}h(x_{n+1}) = y_nw'_nh(x_{n+1}) = h(x_{n+1})\#u_{end}$. Consequently, v_{n+1} and $\#u_{end}$ are conjugate, so $\tilde{v}_{n+1} = \#u_{end}$. Moreover, by Condition (2) of Lemma 10, $v_1 = \tilde{v}_1 = \#u_{start}$. Thus, it follows from Claim 13 that μ satisfies REP. ◀

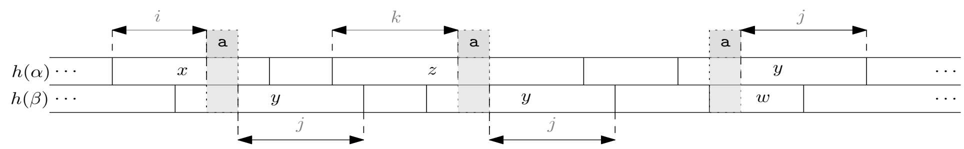

Figure 2 Fixing positions: since an occurrence of the i^{th} letter of $h(x)$ corresponds to an occurrence of the $(|h(y)| - j)^{th}$ letter of y, whose other occurrences correspond to the k^{th} letter of $h(z)$ and first letter of $h(w)$, all these positions are equivalent and contain the same letter, e.g., a.

It is clear that the equation $\alpha_\mu = \beta_\mu$ (and hence also the equation $\alpha = \beta$ given in Lemma 9) may be constructed in polynomial time, therefore our reduction from REP is complete. So, by Lemmas 7 and 11, we have shown Theorem 4.

4 NP-upper bound

In this section, we discuss a series of results related to the satisfiability of regular-ordered word equations. To this end, we extend the classical approach of filling the positions (see e.g., [15] and the references therein). This method essentially comprises of assuming that for a given equation $\alpha = \beta$, we have a solution h with specified lengths $|h(x)|$ for each variable x. The assumption that h satisfies the equation induces an equivalence relation on the positions of each $h(x)$: if a certain position in the solution-word is produced by an occurrence of the i^{th} letter of $h(x)$ on the RHS and an occurrence of the j^{th} letter of $h(y)$ on the LHS, then these two positions must obviously have the same value/letter and we shall say that these occurrences *correspond*. These individual equivalences can be combined to form equivalence classes, and if no contradictions occur (i.e., two different terminal symbols a and b do not belong to the same class), a valid solution can be derived.

Such an approach already allows for some straightforward observations regarding the (non-)minimality of a solution h. In particular, if an equivalence class of positions is not associated with any terminal symbol, then all positions in this class can be mapped to ε, resulting in a strictly shorter solution. On the other hand, even for our restricted setting, this observation is insufficient to provide a bound on the length of minimal solutions. In fact, in the construction of the equivalence classes we ignore, or at least hide, some of the structural information about the solution. In what follows, we shall see that by considering the exact 'order' in which positions are equated, we are able to give some more general conditions under which a solution is not minimal.

For our approach, rather than just constructing these equivalence classes, we shall construct sequences of equivalent positions, and then analyse "similar" sequences. For example, one occurrence of a position i in $h(x)$ might correspond to an occurrence of position j in $h(y)$, while another occurrence of position j in $h(y)$ might correspond to position k in $h(z)$, and so on, in which case we would consider the sequence: $\ldots \to (x, i) \to (y, j) \to (z, k) \to \ldots$.

The sequence terminates when either a variable which occurs only once or a terminal symbol is reached. For general equations, considering all such sequences leads naturally to a graph structure where the nodes are positions $(x, i) \in X \times \mathbb{N}$, and number of edges from each node is determined by the number of occurrences of the associated variable. Each connected component of such a graph corresponds to an equivalence class of positions as before. In the case of quadratic (and therefore also regular) equations, where each variable occurs at most twice, each 'node' (x, i) has at most two edges, and hence our graph is simply a set of disjoint chains, without any loops. As before, each chain (called in the following sequence) must be

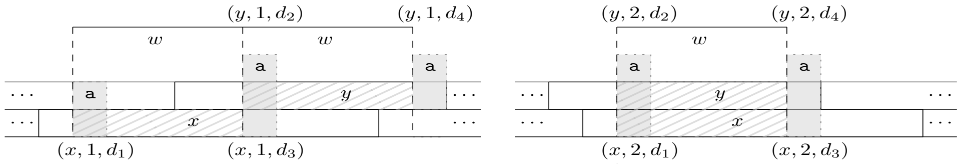

Figure 3 Illustration of Lemma 14 in the case of a short subsequence $\ldots,(x,1,d_1),(y,2,d_2),(x,1,d_3),(y,2,d_4),\ldots$: since the two sequences starting at $(x,1,d_1)$ and $(x,1,d_3)$ are similar, they define a common region w (shaded). Since they are consecutive, the first and last occurrences of w are adjacent, and on opposite sides of the equation. Thus, removing the region w from $h(x)$ and $h(y)$ does not alter the fact that h satisfies the equation.

associated with some occurrence of a terminal symbol, which must occur either at the start or the end. Hence we have $k < n$ sequences, where n is the length of the equation, such that every position (x,i), where x is a variable and $1 \leq i \leq |h(x)|$, occurs in exactly one sequence. It is also not hard to see that the total length of the sequences is upper bounded by $2|h(\alpha)|$.

In order to be fully precise, we will distinguish between different occurrences of a variable/terminal symbol by associating each with an index $z \in \mathbb{N}$ by enumerating occurrences from left to right in $\alpha\beta$. When considering quadratic equations, $z \in \{1,2\}$ for each variable x.

Formally, we define our sequences for a given solution h to a quadratic equation $\alpha = \beta$ as follows: a *position* is a tuple (x,z,d) such that x is a variable or terminal symbol occurring in $\alpha\beta$, $1 \leq z \leq |\alpha\beta|_x$, and $1 \leq d \leq |h(x)|$. Two positions (x,z,d) and (y,z',d') *correspond* if they generate the same position in the solution-word. The positions are *similar* if they belong to the same occurrence of the same variable (i.e., $x = y$ and $z = z'$). For each position p associated with either a terminal symbol or a variable occurring only once in $\alpha\beta$, we construct a sequence $S_p = p_1, p_2, \ldots$ such that

- $p_1 = p$ and p_2 is the (unique) position corresponding with p_1, and
- for $i \geq 2$, if $p_i = (x,z,d)$ such that x is a terminal symbol or occurs only once in $\alpha\beta$, then the sequence terminates, and
- for $i \geq 2$, if $p_i = (x,z,d)$, such that x is a variable occurring twice, then p_{i+1} is the position corresponding to the (unique) position (x,z',d) with $z' \neq z$ (i.e., the 'other' occurrence of the i^{th} letter in $h(x)$).

We extend the idea of similarity from positions to sequences of positions in the natural way: two sequences $p_1,p_2,\ldots,p_i$ and $q_1,q_2,\ldots,q_i$ are *similar* whenever p_j and q_j are similar for all $j \in \{1,2,\ldots,i\}$. Our main tool is the following lemma, which essentially shows that if a sequence contains two similar consecutive subsequences, then the solution defining that sequence is not minimal.

▶ **Lemma 14.** *Let h be a solution to a quadratic equation $\alpha = \beta$, and let p be a position associated with a single-occurring variable or terminal symbol. If the sequence S_p has a subsequence $p_1,p_2,\ldots,p_t,p_{t+1},p_{t+2},\ldots,p_{2t}$ such that $p_1,p_2,\ldots,p_t$ and $p_{t+1},p_{t+2},\ldots,p_{2t}$ are similar, then h is not minimal.*

Proof. Assume that S_p has such a subsequence and assume w.l.o.g. that it is length-minimal (so t is chosen to be as small as possible). For $1 \leq i \leq 2t$, let $p_i = (x_i,z_i,d_i)$ and note that by definition of similarity, for $1 \leq i \leq t$, $x_i = x_{i+t}$ and $z_i = z_{i+t}$. Assume that $d_1 < d_{t+1}$ (the case where $d_1 > d_{t+1}$ may be treated identically). We need the following two claims, whose proofs are omitted due to space constraints.

▶ **Claim 15.** *Suppose that* $(x, z, d), (x, z, d'), (x, z, d'')$ *are positions with* $d \leq d' < d''$ *such that* (x, z, d) *and* (x, z, d'') *correspond to* (y, z', e) *and* (y, z', e'') *respectively. Then* $e'' - e = d'' - d$*, and there exists* e' *with* $e' - e = d' - d$ *such that* (x, z, d') *and* (y, z', e') *correspond.*

A straightforward consequence of Claim 1 is that there exists a constant $C \in \mathbb{N}$ such that for all $i \in \{1, 2, \ldots, t\}$, $d_{i+t} - d_i = C$. Intuitively, each pair of similar positions $p_i = (x_i, z_i, d_i)$ and $p_{i+t} = (x_i, z_i, d_i + C)$ are the first and last positions of a factor $h(x_i)[d_i..d_i + C - 1]$, which as we shall see later on in the proof, can be removed to produce a shorter solution g.

It also follows from Claim 1, along with the fact that we chose t to be minimal, that there does not exist a position $p_i = (x_1, z_1, k)$ in the chain such that $d_1 < k < d_1 + C$. Symmetrically, there does not exist a position $p_i = (x_t, z_t, k)$ such that $d_t < k < d_t + C$:

▶ **Claim 16.** *Let* $j \in \{2, \ldots, t, t + 2, \ldots, 2t\}$ *such that* $x_j = x_1 (= x_{t+1})$ *and* $z_j = z_1 (= z_{t+1})$*. Then* $d_j \notin \{d_1, \ldots, d_{t+1} (= d_1 + C)\}$*. Likewise, if* $j \in \{1, \ldots, t - 1, t + 1, \ldots, 2t - 1\}$ *such that* $x_j = x_t$ *and* $z_j = z_t$*, then* $d_j \notin \{d_t, \ldots, d_{2t}\}$*.*

Using the observations above, we shall remove parts of the solution h to obtain a new, strictly shorter solution and thus show that h is not minimal as required.

To do this, we shall define a new equation $\alpha' = \beta'$ obtained by replacing the second occurrence of each variable x (when it exists) with a new variable x'. The reason is so we may delete factors independently from different occurrences of $h(x_i)$, and more easily keep track of the equivalence of the left and right hand side of the equation. Note that we can derive a solution h' to $\alpha' = \beta'$ from the solution h to our original equation by simply setting $h'(x) = h'(x') = h(x)$ for all $x \in \mathsf{var}(\alpha\beta)$. Likewise, any solution to $\alpha' = \beta'$ for which this condition holds (i.e., $h'(x) = h'(x')$ for all $x \in \mathsf{var}(\alpha\beta)$) induces a solution g to our original equation $\alpha = \beta$ given by $g(x) = h'(x)(= h'(x'))$. Finally, for each position (x, z, d) in the original solution h, there exists a unique "associated position" in h' given by $h(x)[d]$ if $z = 1$ and $h(x')[d]$ if $z = 2$. Furthermore, it follows from the definitions that for any pair of positions p, q which correspond (in terms of h), we can remove the associated positions from h' and the result will still be a valid solution to our modified equation $\alpha' = \beta'$ (although such a solution may no longer induce a valid solution to our original equation, since it is no longer necessarily the case where $h'(x) = h'(x')$ for all x).

We construct our shorter solution g to $\alpha = \beta$ as follows. Let h' be the solution to $\alpha' = \beta'$ derived directly from h. Recall from the definition of S_p that, for $1 \leq i < t$, the positions $(x_i, \overline{z}_i, d_i)$ and $(x_{i+1}, z_{i+1}, d_{i+1})$ correspond, where $\overline{z} = (z + 1) \mod 2$ (i.e., so that $\overline{z} \neq z$). Moreover, $(x_i, \overline{z}_i, d_i + C)$ and $(x_{i+1}, z_{i+1}, d_{i+1} + C)$ correspond, and thus by Claim 1, $(x_i, \overline{z}_i, d_i + k)$ and $(x_{i+1}, z_{i+1}, d_{i+1} + k)$ correspond for $0 \leq k \leq C - 1$. Since corresponding positions must have the same value/letter, it follows that there exists a factor $w \in \Sigma^+$ such that $w = h(x_i)[d_i..d_i + C - 1](= h'(x)[d_i..d_i + C - 1] = h'(x')[d_i..d_i + C - 1])$ for $1 \leq i \leq t$.

We now produce a new solution, h'' to $\alpha' = \beta'$ as follows. For each i such that $1 \leq i \leq t-1$ and for each k such that $d_i \leq k \leq d_i + C - 1$, delete from h' the pair of positions associated with $(x_i, \overline{z}_i, d_i + k)$ and $(x_{i+1}, z_{i+1}, d_{i+1} + k)$. Note that since these positions correspond, in deleting them, we continue to have a valid solution to $\alpha' = \beta'$. Notice also that for every position associated with $(x_i, \overline{z}_i, d_i + k)$ such that $1 < i \leq t$, we also delete the position associated with $(x_i, z_i, d_i + k)$. Hence, for all $x \notin \{x_1, x_t\}$, $h''(x) = h''(x')$. The same does not necessarily hold for x_1 and x_t, however. In particular, we deleted positions associated with $(x_1, \overline{z_1}, d_1 + k)$ and $(x_t, z_t, d_t + k)$ for $0 \leq k \leq C - 1$, but not their counterparts $(x_1, z_1, d_1 + k)$ and $(x_t, \overline{z}_t, d_t + k)$. Hence, to derive a valid solution g, we must also delete these positions. To see that, in doing so, we still have a valid solution to $\alpha' = \beta'$, note firstly that, by Claim 2, we have not deleted any of these positions already. Moreover, it follows from the sequence S_p that $(x_t, \overline{z}_t, d_t)$ corresponds to $(x_1, z_1, d_1 + C)$. Assume $z_1 = 1$ (the case $z_1 = 2$ is symmetric). It follows that $z_t = 1$ (since $\overline{z}_t \neq z_1$). Thus there exists an index m such that $h''(x_1)[d_1..d_1 + C - 1]$ generates the factor w starting at position m in $h''(\alpha')$ and $h''(x_t)[d_t..d_t + C - 1]$ generates the (same) factor w starting at position $m + |w|$ in $h''(\beta)$. It is straightforward to see that removing these factors (i.e., deleting the positions associated with $(x_1, z_1, d_1 + k)$ and $(x_t, \overline{z}_t, d_t + k)$ for $0 \leq k \leq C - 1$) does not affect the agreement of the two sides of the equation. Thus we obtain a shorter solution h'' to $\alpha' = \beta'$ such that $h(x) = h(x')$ for all variables x, hence a shorter solution g given by $g(x) = h''(x)$ to $\alpha = \beta$. ◀

An important fact related to the representation of the minimal solutions of quadratic equations as oriented sequences of positions, that start or end with a terminal symbol, is that the number of such sequences is linear in the size of the equation. Also, for variables that occur only once in a quadratic equation (so, only in one side), each of their positions is the start or end of a chain, so their length can be at most linear in the size of the equation. Thus, when interested in showing that a certain class of quadratic equation is in NP, we can assume that these equations do not contain variables occurring only once. Such variables could be non-deterministically replaced by words of at most linear size, and we would have to solve a new equation, whose size is at most quadratic in the size of the original one.

Further, using Lemma 14 and the observations above, we obtain immediately that minimal solutions to regular-ordered equations are at most linear in the length of the equation.

▶ **Proposition 17.** *Let E be a regular-ordered word equation with length n, and let h be a minimal solution to E. Then $|h(x)| < n$ for each variable x occurring in E.*

Proof. Every position of a minimal solution h to E occurs somewhere in one of the associated sequences S_p, and there are no more than n such sequences. So, it is sufficient to show that each one contains at most one position (x, z, d) for each variable x. This follows easily from the fact that E is regular and ordered and h is minimal, so it fulfils the restrictions of Lemma 14. ◀

We can see that, in terms of restricting the lengths of individual variables, the result in Proposition 17 is optimal. For instance, in a minimal solution h to the equation $w\mathtt{c}x_1 = x_1\mathtt{c}w$, with $w \in \{a, b\}^*$, the variable x_1 is mapped to w, so $|h(x)| = |E| - 2 \in O(|E|)$. Furthermore, Theorem 18 follows now as a direct consequence of Proposition 17 and Lemma 1, as the length of a minimal solution to a regular-ordered equation $\alpha = \beta$ is $O(|\alpha\beta|^2)$.

▶ **Theorem 18.** *The satisfiability problem for regular-ordered equations is in* NP*.*

It is a simple consequence of Proposition 17 that the satisfiability of a regular-ordered equation E with a constant number k of variables can be checked in P-time.

We now consider the satisfiability problem for regular equations with regular constraints, where we can make further use of the ideas presented so far in this section. Firstly, we observe that the problem is, in general, PSPACE-complete, even for regular-ordered equations (the proof given in [21] that the problem is PSPACE-complete for regular equations is also sufficient for the regular-ordered case).

▶ **Proposition 19.** *The satisfiability of regular-ordered equations with regular constraints is* PSPACE*-complete*

However, when considering regular equations whose sides contain exactly the same variables, and, moreover, in the same order, we get the following result, which shows an important difference between the unrestricted regular-ordered equations, where the variables occurring only on one side do not affect the complexity of the satisfiability problem, and the same type of equations subject to regular constraints.

▶ **Theorem 20.** *The satisfiability of regular-ordered equations whose sides contain exactly the same variables, with regular constraints, is in* NP*.*

The same upper bound holds also for regular-ordered equations with regular constraints, when the regular constraints are regular languages accepted by NFAs with at most c states, where c is a constant (called constant regular constraints in the following).

▶ **Theorem 21.** *The satisfiability problem for regular-ordered equations with constant regular constraints is in* NP*.*

The proofs of the last two results rely also on the approach outlined in Lemma 14, but they are more involved. We first define the sequences of positions fixed by the terminals in a minimal solution of an equation. While this time the sequences may contain repetitions, we show that they cannot contain repetitions of non-constant exponent (where the constant, however, depends exponentially on c). This provides, again, a polynomial upper bound on the length of minimal solutions of such equations.

5 Tractable equations

Finally, we discuss a class of equations for which satisfiability is in P. Tractability was obtained so far from two sources: bound the number of variables by a constant (e.g., one or two-variable equations [13, 1]), or heavily restrict their structure (e.g., regular equations whose sides do not have common variable, or equations that only have one repeated variable, but at least one non-repeated variable on each side [21]). The class we consider slightly relaxes the previous restrictions. As the satisfiability of quadratic or even regular-ordered equations is NP-hard it seems reasonable to consider here patterns where the number of repeated variables is bounded by a constant (but may have an arbitrary number of non-repeated variables). More precisely, we consider here non-cross equations with only one repeated variable. This class generalises naturally the class of one-repeated variables.

▶ **Theorem 22.** *Let $x \in X$ be a variable and $\mathcal{D}$ be the class of word equations $\alpha = \beta$ such that $\alpha, \beta \in (\Sigma \cup X)^*$ are non-cross and each variable of X other than x occurs at most once in $\alpha\beta$. Then the satisfiability problem for $\mathcal{D}$ is in* P*.*

In the light of the results from [21], it follows that the interesting case of the above theorem is when the equation $\alpha = \beta$ is such that $\alpha = xu_1xu_2 \cdots u_kx$ and $\beta = \beta' v_0xv_1xv_2 \cdots xv_k\beta''$ where $v_0, v_1, \ldots, v_k, u_1, u_2, \ldots, u_k \in \Sigma^*$ and β', β'' are regular patterns that do not contain x and are variable disjoint. Essentially, this is a matching problem in which we try to align two non-cross patterns, one that only contains a repeated variable and constants, while the other contains the repeated variable, constants, and some wild-cards that can match any factor. As the proofs of [21] used essentially the non-repeated variables occurring in each of the sides, a novel and much more involved approach was needed here. The idea of our proof is to first show that such equations have minimal solutions of polynomial length. Further, we note that if we know the length of β' (w.r.t. the length of α) then we can determine the position where the factor $v_0xv_1xv_2 \cdots xv_k$ occurs in α, so the problem boils down to seeing how the positions of x are fixed by the constant factors v_i. Once this is done, we check if there exists an assignment of the variables of β' and β'' such that the constant factors of these patterns fit correctly to the corresponding prefix, respectively, suffix of α.

6 Conclusions and Prospects

The main result of this paper is the NP-completeness of the satisfiability problem for regular-ordered equations. While the lower bound seems remarkable to us because it shows that solving very simple equations, which also always have short solutions, is NP-hard, the upper bound seems more interesting from the point of view of the tools we developed to show it. We expect the combinatorial analysis of sequences of equivalent positions in a minimal solution to an equation (which culminated here in Lemma 14) can be applied to obtain upper bounds on the length of the minimal solutions to more general equations than just the regular-ordered ones. It would be interesting to see whether this type of reasoning leads to polynomial upper bounds on the length of minimal solutions to regular (not ordered) or quadratic equations, or to exponential upper bounds on the length of minimal solutions of non-cross or cubic equations. In the latter cases, a more general approach should be used, as the equivalent positions can no longer be represented as linear sequences, but rather as directed graphs.

Regarding the final section our paper, it seems interesting to us to see whether deciding the satisfiability of word equations with one repeated variable (so without the non-cross sides restriction) is still tractable. Also, it seems interesting to analyse the complexity of word equations where the number of repeated variables is bounded by a constant.

References

1. R. Dąbrowski and W. Plandowski. Solving two-variable word equations. In *Proc. 31th International Colloquium on Automata, Languages and Programming, ICALP 2004*, volume 3142 of *Lecture Notes in Computer Science*, pages 408–419, 2004.
2. V. Diekert, A. Jez, and M. Kufleitner. Solutions of word equations over partially commutative structures. In *Proc. 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 127:1–127:14, 2016.
3. V. Diekert and J. M. Robson. On quadratic word equations. In *Proc. 16th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1999*, volume 1563 of *Lecture Notes in Computer Science*, pages 217–226, 1999.
4. A. Ehrenfeucht and G. Rozenberg. Finding a homomorphism between two words is NP-complete. *Information Processing Letters*, 9:86–88, 1979.
5. H. Fernau, F. Manea, R. Mercaş, and M.L. Schmid. Pattern matching with variables: Fast algorithms and new hardness results. In *Proc. 32nd Symposium on Theoretical Aspects of Computer Science, STACS 2015*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 302–315, 2015.
6. H. Fernau and M. L. Schmid. Pattern matching with variables: A multivariate complexity analysis. *Information and Computation*, 242:287–305, 2015.
7. H. Fernau, M. L. Schmid, and Y. Villanger. On the parameterised complexity of string morphism problems. *Theory of Computing Systems*, 2015. http://dx.doi.org/10.1007/s00224-015-9635-3.
8. D. D. Freydenberger. A logic for document spanners. In *Proc. 20th International Conference on Database Theory, ICDT 2017*, Leibniz International Proceedings in Informatics (LIPIcs), 2017. To appear.
9. D. D. Freydenberger and M. Holldack. Document spanners: From expressive power to decision problems. In *Proc. 19th International Conference on Database Theory, ICDT 2016*, volume 48 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:17, 2016.

10 M. R. Garey and D. S. Johnson. *Computers And Intractability.* W. H. Freeman and Company, 1979.
11 J. Jaffar. Minimal and complete word unification. *Journal of the ACM*, 37(1):47–85, 1990.
12 A. Jez. Context unification is in PSPACE. In *Proc. 41st International Colloquium on Automata, Languages, and Programming, ICALP 2014*, volume 8573 of *Lecture Notes in Computer Science*, pages 244–255. Springer, 2014.
13 A. Jeż. One-variable word equations in linear time. *Algorithmica*, 74:1–48, 2016.
14 A. Jeż. Recompression: A simple and powerful technique for word equations. *Journal of the ACM*, 63, 2016.
15 J. Karhumäki, F. Mignosi, and W. Plandowski. The expressibility of languages and relations by word equations. *Journal of the ACM*, 47:483–505, 2000.
16 A. Koscielski and L. Pacholski. Complexity of Makanin's algorithm. *Journal of the ACM*, 43(4):670–684, 1996.
17 M. Lothaire. *Algebraic Combinatorics on Words.* Cambridge University Press, Cambridge, New York, 2002.
18 R. C. Lyndon. Equations in free groups. *Transactions of the American Mathematical Society*, 96:445–457, 1960.
19 R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.
20 G. S. Makanin. The problem of solvability of equations in a free semigroup. *Matematicheskii Sbornik*, 103:147–236, 1977.
21 F. Manea, D. Nowotka, and M. L. Schmid. On the solvability problem for restricted classes of word equations. In *Proc. 20th International Conference on Developments in Language Theory, DLT 2016*, volume 9840 of *Lecture Notes in Computer Science*, pages 306–318. Springer, 2016.
22 W. Plandowski. An efficient algorithm for solving word equations. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, STOC 2006*, pages 467–476, 2006.
23 W. Plandowski and W. Rytter. Application of Lempel-Ziv encodings to the solution of words equations. In *Proc. 25th International Colloquium on Automata, Languages and Programming, ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 731–742. Springer, 1998.
24 D. Reidenbach and M. L. Schmid. Patterns with bounded treewidth. *Information and Computation*, 239:87–99, 2014.
25 K. U. Schulz. Word unification and transformation of generalized equations. *Journal of Automated Reasoning*, 11:149–184, 1995.