Solutions of Twisted Word Equations, EDT0L Languages, and Context-Free Groups*†

Volker Diekert[1] and Murray Elder[2]

1 **Universität Stuttgart, Formal Methods in CS, Stuttgart, Germany**
diekert@fmi.uni-stuttgart.de
2 **University of Technology Sydney, Sydney, Australia**
murray.elder@uts.edu.au

Abstract

We prove that the full solution set of a twisted word equation with regular constraints is an EDT0L language. It follows that the set of solutions to equations with rational constraints in a context-free group (= finitely generated virtually free group) in reduced normal forms is EDT0L. We can also decide whether or not the solution set is finite, which was an open problem. Moreover, this can all be done in PSPACE. Our results generalize the work by Lohrey and Sénizergues (ICALP 2006) and Dahmani and Guirardel (J. of Topology 2010) with respect to complexity and with respect to expressive power. Both papers show that satisfiability is decidable, but neither gave any concrete complexity bound. Our results concern all solutions, and give, in some sense, the "optimal" formal language characterization.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, F.4.2 Grammars and Other Rewriting Systems, F.4.3 Formal Languages

Keywords and phrases Twisted word equation, EDT0L, virtually free group, context-free group

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.96

1 Introduction

In a seminal paper [21] Makanin showed that the problem *WordEquations* is decidable. The first complexity estimation of that problem was a tower of several exponential functions, but this dropped down to PSPACE by Plandowski [24] using compression. The insight that long solutions of word equations can be efficiently compressed is due to [25], which led to the conjecture that WordEquations is NP-complete. In 2013 Jeż applied his *recompression* technique: he presented a new and simple NSPACE($n \log n$) algorithm to solve word equations [16]. (Very recently, he lowered the complexity to NSPACE(n) [17]). Actually his method achieved more: it describes all solutions, copes with rational constraints (which is essential in applications), and it extends to free groups [6]. Building on ideas in [6], Ciobanu and the present authors showed that the full solution set of a given word equation with rational constraints is EDT0L [3]. This was known before only for quadratic word equations by [11]. *EDT0L-languages* are defined by a certain type of Lindenmayer system, see [27]. The motivation for [3] was to prove that the full solution set in reduced words of equations in free groups is an indexed language, an open problem at the time [12, 15]. But EDT0L is better: it is strictly included in the class of indexed languages [9].

* A full version of the paper is available at https://arxiv.org/abs/1701.03297.
† Research supported by Australian Research Council (ARC) Project DP 160100486 and German Research Foundation (DFG) Project DI 435/7-1

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 96; pp. 96:1–96:14


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The class of finitely generated (f.g. for short) virtually free groups arises in many different ways. A fundamental theorem of Muller and Schupp (relying on [8]) says that a f.g. group is virtually free if and only if it is *context-free* [23]. This means that, given any set of monoid generators A, the set of words $w \in A^*$ which represent $1 \in V$ forms a context-free language. Other characterizations include: (1) fundamental groups of finite graphs of finite groups [18], (2) f.g. groups having a Cayley graph with finite treewidth [19], (3) groups having a finite presentation by some geodesic string rewriting system [13], and (4) f.g. groups having a Cayley graph with decidable monadic second-order theory [19], etc. See [7]. We show that given a f.g. virtually free group V there is a PSPACE-algorithm which produces, for a given equation with rational constraints, an EDT0L grammar which describes the full solution set in reduced words over a natural set of generators. Several remarks are in order here. First, virtually free groups (which are not free) have torsion, and this is serious obstacle to applying the techniques used in [24, 16, 6, 3]. A driving motivation to study virtually free groups is the connection to *word hyperbolic groups* [14]. Solving equations in torsion-free hyperbolic groups reduces to solving equations in free groups [26], but solving equations in word hyperbolic groups with torsion reduces to solving equations in virtually free groups which in turn reduces to solving "twisted" word equations with rational constraints [4]. The question how to solve "twisted" word equations was asked by Makanin ([22, Problem 10.26(b)]) and solved by Lohrey and Sénizergues [20] and Dahmani and Guirardel [4]. Both papers show more general results, and yield independent proofs that satisfiability for equations over a f.g. virtually free group is decidable. The approach in [4] assumes a bound on the so-called "exponent of periodicity", thus it does not handle the full set of solutions. Lohrey and Sénizergues [20] prove a general transfer result which applies to all solutions, but this does not produce any "nice" description. Note that to have "some description" of all solutions is not enough to decide finiteness, in general. Our EDT0L description pays attention that every solution is represented exactly once. The other achievement here is a first known concrete complexity bound: PSPACE, a surprisingly low complexity given the circumstances.

Therefore, the present paper extends [4, 20] in various aspects. As in [4] we are working over twisted word equations with rational constraints, which is the natural approach due to Bass-Serre theory [31], see [18] (and [29, 30] for effective constructions). Our main new contribution is within combinatorics on words. Although we follow the general scheme [16, 6, 3] to define a sound and complete algorithm to produce an NFA describing all solutions, the technical details are quite far from previous methods.

Proofs omitted from the present paper can be found in [5].

1.1 Preliminaries

An *alphabet* is a finite set of *letters*; and Σ^* denotes the free monoid of *words* over Σ. The empty word is denoted by 1. The length $w \in \Sigma^*$ is $|w|$, and $|w|_a$ counts how often a letter a appears in w. Let M be any monoid. Then $u \in M$ is a *factor* of $v \in M$ if we can write $v = xuy$ for some x, y. If $x = 1$ (resp. $y = 1$), then we say that u is a *prefix* (resp. *suffix*) of v. For a prefix, we also write $u \leq v$. An *involution* is a bijection $x \mapsto \overline{x}$ such that $\overline{\overline{x}} = x$ for all x in the set. A *monoid with involution* additionally has to satisfy $\overline{xy} = \overline{y}\,\overline{x}$. If G is a group, then it is a monoid with involution by taking $\overline{g} = g^{-1}$ for all $g \in G$. Thus, we identify $\overline{g}$ and g^{-1} in groups. In the following, every alphabet comes with an involution. This is no restriction since the identity is always an involution for sets. A *morphism* between sets with involution is a mapping respecting the involution. A *morphism* between monoids with involution is a homomorphism $\varphi : M \to N$ such that $\varphi(\overline{x}) = \overline{\varphi(x)}$. For $\Delta \subseteq M \cap N$ we say that it is a Δ-*morphism* if $\varphi(x) = x$ for all $x \in \Delta$. A bijective morphism from a set to itself is called an *automorphism* and the set of automorphisms on a set (or monoid)

M forms the group $\mathrm{Aut}(M)$. Let G be a group. It acts on a set (with involution) X by a mapping $x \mapsto g \cdot x$ if $1 \cdot x = x$, $f \cdot (g \cdot x) = (fg) \cdot x$ (and $f \cdot \overline{x} = \overline{f \cdot x}$). If G acts on a monoid (with involution) M, then we additionally demand that every group element acts as an automorphism: $f \cdot (xy) = (f \cdot x)(f \cdot y)$. Frequently, we write $f(x)$ instead of $f \cdot x$. The specification of regular constraints is given here by assigning to each constant and variable an element in a finite monoid (typically the finite monoid is a monoid of Boolean matrices and arises as the transformation monoid of a finite automaton.) By making the finite monoid larger, we can turn it into a monoid N with involution and where G acts on it. This allows us to represent regular constraints using a morphism $\mu : (A \cup (G \times \mathcal{X}))^* \to N$ which respects the involution and the action of G. In the following we fix the finite monoid N and we assume that all morphisms to N respect the involution and G action. We say that M is an *NG-i-monoid* if M is a monoid with involution and a G action together with a morphism $\mu : M \to N$. (In this abbreviation the i stands for "involution".) If not explicitly stated otherwise all monoids under consideration are NG-i-monoids (including N itself). A morphism between NG-i-monoids M, M' with morphisms μ, μ' is a morphism $\varphi : M \to M'$ such that $\varphi(g \cdot x) = g \cdot (\varphi(x))$ and $\mu'\varphi = \mu$. Henceforth, by default, a morphism means a morphism between NG-i-monoids.

Regular languages in finitely generated free monoids can be defined via nondeterministic finite automata (NFA for short) or via recognizability via homomorphisms to finite monoids, to mention just two possible definitions. This notion extends to every monoid M: an NFA is a directed finite graph $\mathcal{A}$ with initial and final *states*, where the transitions are labeled with elements of the monoid M. A transition labeled by $1 \in M$ is called an *ε-transition*. We say that $m \in M$ is *accepted* by the automaton $\mathcal{A}$ if there exists a path from some initial to some final state such that multiplying the edge labels together yields m. This defines the accepted language $L(\mathcal{A}) = \{m \in M \mid m \text{ is accepted by } \mathcal{A}\}$. According to [10] a subset $L \subseteq M$ is *rational* if and only if L is accepted by some NFA over M. An NFA is called *trim* if every state is on some path from an initial to a final state. Ensuring the NFA that we construct in our proof below is trim, allows us to decide emptiness or finiteness of the solution set.

A subset $L \subseteq A^* \times \cdots \times A^*$ is called *EDT0L* if there some (extended) alphabet C with $c_1, \ldots, c_k \in C$ such that $A \subseteq C$ and a rational set $\mathcal{R} \subseteq \mathrm{End}(C^*)$ of endomorphisms over C^* such that $L = \{(h(c_1), \ldots, h(c_k)) \mid h \in \mathcal{R}\}$. The classical definition for EDT0L refers to $k = 1$. Our definition uses a characterization of EDT0L languages due to Asveld [1, 28].

Let B and $\mathcal{Y}$ be two disjoint NG-i-alphabets. We call B the alphabet of *constants* and $\mathcal{Y}$ the set of *twisted variables*. It is convenient to choose a minimal subset $\mathcal{X} \subseteq \mathcal{Y}$ such that every $Y \in \mathcal{Y}$ has the form $Y = f \cdot X$ for some $X \in \mathcal{X}$ and $f \in G$. Moreover, we assume $X \neq \overline{X}$ for all variables. If G acts without fixed points on $\mathcal{Y}$, then we identify $\mathcal{Y} = G \times \mathcal{X}$ and the action becomes $g \cdot (f, X) = (gf, X)$. By $M(B, \mathcal{X}, \theta, \mu)$ we denote an NG-i-monoid which is generated by $B \cup \{f(X) \mid f \in G, X \in \mathcal{X}\}$ together with a finite set θ of *homogeneous* defining relations, which means every $(x, y) \in \theta$ satisfies $|x| = |y|$. We always assume that $(x, y) \in \theta$ implies $\mu(x) = \mu(y)$, $(\overline{x}, \overline{y}) \in \theta$, and $(f(x), f(y)) \in \theta$ for all $f \in G$, even if these relations are not listed in the specification of θ. For complexity issues we require $|x| \leq 2$ for each $(x, y) \in \theta$ and $|\theta| \in \mathcal{O}(|G| \, \|\mathcal{S}\|^2)$ where $\|\mathcal{S}\|$ is specified in Theorem 1. The homogeneity condition makes it possible to solve the word problem and all other computational issues for the quotient $M(B, \mathcal{X}, \theta, \mu) = M(B, \mathcal{X}, \emptyset, \mu)/\{x = y \mid (x, y) \in \theta\}$ within our space bound.

2 The main results

Let A an alphabet of constants and G be a subgroup of $\mathrm{Aut}(A)$. Initially, the set of twisted variables is $G \times \mathcal{V}$. For a word $w \in A^*$ and $f \in G$ we use the notation $f(w) = (f, w)$;

and we hence identify $(A \cup (G \times \mathcal{V}))^* = ((G \times (A^* \cup \mathcal{V}))^*$. We abbreviate $(1, x)$ as x for $x \in A^* \cup \mathcal{V}$. A system $\mathcal{S}$ of *twisted word equations with rational constraints* is given by a set of pairs $\{(U_i, V_i) \mid 1 \leq i \leq s\}$ where $U_i, V_i \in (A \cup (G \times \mathcal{V}))^*$ are *twisted words* and a morphism $\mu_0 : (A \cup (G \times \mathcal{V}))^* \to N$. It is specified by its restriction to $A \cup \mathcal{V}$; and μ_0 respects the involution and the action of G.

As usual, a twisted equation (U_i, V_i) is also written as $U_i = V_i$. A *solution* of $\mathcal{S}$ is given a morphism $\sigma : \mathcal{V} \to A^*$ which is (uniquely) extended to an A-morphism of NG-i-monoids $\sigma : (A \cup (G \times \mathcal{V}))^* \to A^*$ such that $\sigma(U_i) = \sigma(V_i)$ for all i and $\mu_0\sigma(X) = \mu_0(X)$ for all variables. Hence, $\mu_0\sigma = \mu_0$. The full *solution set* $\mathrm{Sol}(\mathcal{S})$ for $\mathcal{V} = \left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}\right\}$ is $\mathrm{Sol}(\mathcal{S}) = \{(\sigma(X_1), \ldots, \sigma(X_k)) \in A^* \times \cdots \times A^* \mid \sigma \text{ solves } \mathcal{S}\}$. We define the size $\|\mathcal{S}\|$ by $\|\mathcal{S}\| = |G| + |A| + |\mathcal{V}| + s + \sum_{1 \leq i \leq s} |U_i V_i|$.

Convention. For better readability we don't measure N, but we add the general hypotheses that N is given in such a way that the specification and all necessary computations over N (multiplication, computing the involution and the G action) can be done in polynomial space with respect to $\|\mathcal{S}\|$. This is no restriction, as we can add trivial equations to enlarge $\|\mathcal{S}\|$.

▶ **Theorem 1.** *There is a* PSPACE *algorithm which takes as input a system of twisted word equations with rational constraints $\mathcal{S}$ as above with input size $\|\mathcal{S}\|$. The output is an extended alphabet C of size $\mathcal{O}(|G|^2 \|\mathcal{S}\|^2)$, letters $c_X \in C$ for each $X \in \mathcal{V}$, and a trim NFA $\mathcal{A}$ accepting a rational set of A-morphisms $L(\mathcal{A}) \subseteq \mathrm{End}(C^*)$ such that*

$$\mathrm{Sol}(\mathcal{S}) = \left\{(h(c_{X_1}), \ldots, h(c_{X_{|\mathcal{V}|}})) \in C^* \times \cdots \times C^* \;\middle|\; h \in L(\mathcal{A})\right\}. \tag{1}$$

Intermediate equations, which label states of the NFA, have a length bound in $\mathcal{O}(|G| \|\mathcal{S}\|^2)$. Moreover, $\mathrm{Sol}(\mathcal{S}) = \emptyset$ if and only if $L(\mathcal{A}) = \emptyset$, and $|\mathrm{Sol}(\mathcal{S})| < \infty$ if and only if $\mathcal{A}$ doesn't contain any directed cycle.

The result above is far-reaching extension of Makanin's classical result on pure word equations. It combines combinatorics on words, automata theory, formal languages, and group actions on alphabets. It doesn't use band complexes, Makanin-Razborov diagrams or results from algebraic geometry over groups [4, 2]. Here, a virtually free group V is given by a group extension of a free group $F(B)$ with a finite group G with the natural set $A = B \cup B^{-1} \cup G \setminus \{1\}$ as generators. We represent elements of V by *reduced normal forms* in $\widehat{V}$, where $\widehat{V}$ is the set of words in $B^*G \subseteq A^*$ without factors $b\overline{b}$. Thus, we have a natural notion of *solution in reduced normal forms.*

▶ **Corollary 2.** *Let V be a f.g. virtually free group. There is an* NSPACE$(m^2\|\mathcal{S}\|^2 \log(\|\mathcal{S}\|))$ *algorithm such that:*

Input. *A system $\mathcal{S}$ of s equations $U_i = V_i$ over V with rational constraints and in variables $X_1, \ldots, X_k$, where $\|\mathcal{S}\| = k + \sum_{1 \leq i \leq s} |U_i V_i|$ and m denotes the number of states for the NFA's to encode constraints.*

Output. *An extended alphabet C of size $\mathcal{O}(\|\mathcal{S}\|^2)$, letters $c_X \in C$ for each variable, and a trim NFA $\mathcal{A}$ accepting a rational set of A-morphisms over C^* such that*

$$\left\{(h(c_{X_1}), \ldots, h(c_{X_k})) \in (C^*)^k \;\middle|\; h \in L(\mathcal{A})\right\} = \left\{(\sigma(X_1), \ldots, \sigma(X_k)) \in \widehat{V}^k \;\middle|\; \sigma \text{ solves } \mathcal{S}\right\}.$$

Moreover, there is no solution if and only if $L(\mathcal{A}) = \emptyset$, and there are infinitely many solutions if and only if $\mathcal{A}$ contains a directed cycle.

The reduction of Corollary 2 to Theorem 1 follows [4] very closely, see [5] for details:

1. Embed V into a semi-direct product $F(S) \rtimes G$ using Bass-Serre theory. This encodes $\widehat{V}$ as a rational set in S^*G and allows us to view a system of equations over V (with rational constraints) as a system of twisted word equations with rational constraints over S.
2. Handling of rational constraints by transformations on NFA's by standard methods.
3. Projection of EDT0L languages and respecting reduced normal forms using the fact that the embedding satisfies $B \subseteq S$.

3 Outline of the proof of Theorem 1

The actual proof of Theorem 1 is rather technical, so this extended abstract outlines the central ideas only. The focus is on those parts which are original and where the twisting forces us to deviate from what has been done elsewhere. Jeż's recompression technique is based on two procedures: *block-compression* and *pair-compression*; solutions are obtained by iteratively *popping* the first and last letters of variables (performing moves of the form $X \mapsto aX$), which increases the length of the equation, and *compressing* factors by replacing pairs ab and powers a^λ by a single (new) letter. In the "untwisted" setting, when we compress a pair ab we replace every occurrence of the factor that is "visible" in the equation, but in the twisted case, the pair ab appearing on one side of the equation needs to match with $f(ab)$ on the other side, which causes complications. The basic problem is that twisting of a word $(ab)^\lambda$ by some $f \in G$ may result in $f(ab)^\lambda = (ba)^\lambda$. The complications related to this will become clear below. Therefore we introduce two new procedures. First we define a new and more general δ-*periodic-compression* w.r.t. some $\delta \in \Theta(|G|\,\|\mathcal{S}\|)$. In some sense, δ-periodic-compression removes the problem caused by $f(p)^\lambda = q^\lambda$ where p and q are primitive words of length at most δ. (Powers of long primitive words are then handled in later iterations.) Performing one δ-periodic-compression will result in a situation where "equivalent" positions in the solution are far apart. This property is used for our version of pair-compression without the possibility to "uncross" pairs as is the usual strategy. Instead we do the following. First we pop out from every $\sigma(X)$ rather long prefixes and suffixes. After that we find room to compress enough pairs ab into fresh letters c. We cannot compress pairs by their label (twisting prevents that), so we compress only pairs ab where the corresponding two positions have no equivalent position which is located at some border of an occurrence of a variable. This is our leads to a new definition of *twisted-pair-compression*. Of course, we must define precisely when positions are equivalent and everything must take the action of G and rational constraints into account. Last but not least, we must realize the procedure by following arcs in an NFA where the labels are endomorphisms over some extended alphabet C. This yields the EDT0L property of the full solution set, more importantly it transforms questions about solvability of equations into structural properties of a finite graph.

One of the new features presented here is the ambient algebraic structure: in the case of free monoids (resp. free groups) the intermediate monoids were partially commutative. Twisting leads to more complicated defining relations. More concretely, when working with an equation $U = V$ over constants B and variables $\mathcal{X}$ with constraints defined by an NG-i-monoid morphism $\mu : B \cup \mathcal{X} \to N$, we deal with an NG-i-monoid denoted by $M(B, \mathcal{X}, \theta, \mu)$. The algebraic structure is a quotient monoid $(B \cup \mathcal{X})^* / \{xy = zx \mid (xy, zx) \in \theta\}$, where $x \in B \cup \mathcal{X}$ and $y, z \in (B \cup \mathcal{X})^*$ with $|y| = |z|$. The idea is that, reading a word in $w \in (B \cup \mathcal{X})^*$, the position of x is not fixed, it "floats" by conjugating y to z or vice versa, without changing the length of W. This possibility of "floating" is essential in our approach.

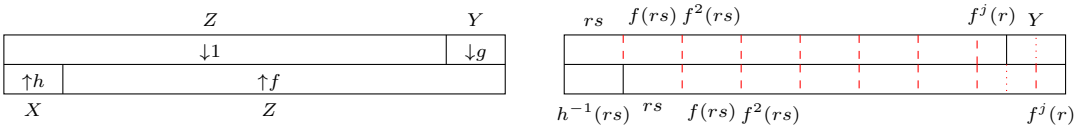


Figure 1 "Graphical" proof of Proposition 3.

3.1 States of the NFA

We start with a system of $\mathcal{S}$ of s equations $U_i = V_i$ over some alphabet A of constants and in variables X_j. We encode $\mathcal{S}$ as a single word using a marker symbol and we obtain the *initial equation* as:

$$W_{\text{init}} = \#X_1\# \cdots \#X_k\#U_1\# \cdots U_s\# \ \overline{\#X_1\# \cdots \#X_k\#V_1\# \cdots V_s\#}. \tag{2}$$

Note that $\sigma(W) = \sigma(\overline{W})$ if and only if $\sigma(U_i) = \sigma(V_i)$ for all i. We fix $n = |W_{\text{init}}|$. Note that this implies $n > |A| + |\mathcal{V}|$ and $\|\mathcal{S}\| \in |G| + \Theta(n)$. States of the transition system are denoted as $(W, B, \mathcal{X}, \theta, \mu)$. We call a state an *extended equation*. Here, B are the current constants and $\mathcal{X}$ are the current variables with $A \subseteq B \subseteq C$ and $\mathcal{X} \subseteq \Omega$ where C and Ω are fixed and of size $\mathcal{O}(|G|^2\|\mathcal{S}\|^2)$ and $W \in M(B, \mathcal{X}, \theta, \mu)$ has length bounded by $\mathcal{O}(|G|\|\mathcal{S}\|^2)$. A solution is a morphism (of NG-i-monoids) $\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ such that $\sigma(W) = \sigma(\overline{W})$. Here, $M(B, \theta, \mu)$ is the submonoid of $M(B, \mathcal{X}, \theta, \mu)$. If θ is empty, then we speak about a *standard state*. We begin at a standard state and the aim is to track for every solution a path from the initial standard state $(W_{\text{init}}, A, G \times \mathcal{V}, \emptyset, \mu_0)$ to some final state $(W, B, \emptyset, \emptyset, \mu)$ without types and variables such that $W = \overline{W}$.

We need to reuse names for constants, so we also need a procedure, called *alphabet-reduction*, to get rid of *invisible* constants. These are letters $b \in B$ where for no $f \in G$ the letter $f(b)$ appears in W. Since a given solution σ might use them, we cannot simply throw them out. This forces us to consider *entire solutions* which are pairs (α, σ) where σ is a solution as above and $\alpha : M(B, \theta, \mu) \to A^*$ is an A-morphism.

3.2 Twisted conjugacy

An important concept in our approach is *twisted conjugacy*. We say that words $x, y \in A^*$ are *twisted conjugate* if there are $f, g, h \in G$ and $z \in A^*$ such that $zg(y) = h(x)f(z)$.

▶ **Proposition 3.** *Let σ be a solution of $Z(g, Y) = (h, X)(f, Z)$ such that $|\sigma(X)|$ satisfies $1 \leq |\sigma(X)| < |\sigma(Z)|$. Then there are words $r \in A^+$, $s \in A^*$ and $e, j \in \mathbb{N}$ with $0 \leq j < |G|$ such that $\sigma(X) = h^{-1}(rs)$ and $\sigma(Z) = ((rs)f(rs) \cdots f^{|G|-1}(rs))^e \, f^0(rs) \cdots f^{j-1}(rs)f^j(r)$.*

3.3 δ-periodic-compression

Recall that $w = a_1 \cdots a_n$ with $a \in A$ has *period* $p \in \mathbb{N}$ if $a_i = a_{i+p}$ for all $1 \leq i \leq n - p$. Let δ be some positive natural number. We say that a word w is *δ-periodic* if it has some period less or equal to than δ. Let u be a prefix (resp. factor, resp. suffix) of some nonempty word w. We say that u is a *maximal δ-periodic prefix (resp. factor, resp. suffix)* in w if we cannot extend the occurrence of the factor u inside w by any letter to the right or left, to get a δ-periodic word. A δ-periodic word w is called *long* if $|w| \geq 3\delta$, and *very long* if $|w| \geq 10\delta$. Standard knowledge in combinatorics on words tells us:

▶ **Lemma 4.** *Let w be a δ-periodic word and $w = p^e r = q^f s$ such that p, q are primitive, $|p| \le |q| \le \delta$, $1 \ne r \le p$, $1 \ne s \le q$, and $|w| \ge 2\delta$. Then $p = q$, $e = f \ge 1$, and $r = s$.*

Let us give a high-level description of our new procedure δ-periodic-compression. For simplicity, we deal just with a single "triangulated" twisted equation $(f, X)w(g, Y) = Z$ where X, Y, Z are variables and $w \in B^*$ is word over the current constants B. We consider a fixed solution σ and we ignore rational constraints by assuming $N = \{1\}$. Moreover, we assume that for every letter $b \in B$ there is some $f \in G$ such that $f(b)$ is a letter in w. Thus, we start with an alphabet-reduction which removes invisible letters for a given solution. Since σ is a solution, we can identify positions in w with positions in $\sigma(Z)$. These identified positions carry the same label and we also say that these positions are *visible*.

Let us consider all very long maximal δ-periodic factors $q^d q'$, written as $up^e rv$, of $\sigma(Z)$ which have an occurrence with a visible position. Note that their total number is bounded by $|w|/\delta$. In the description we assume that $|u| = |v| = 3\delta$, p is primitive of length at most δ and $1 \ne r \le p$. Hence, $up^e rv$ defines the triple (p, r, e) uniquely by Lemma 4.

The idea is that at the end we arrive at a state with a solution where all occurrences of these factors $up^{e_\lambda} rv$ are replaced by $u[r, s, \lambda]v$ where $[r, s, \lambda]$ is the notation for a single fresh letter and $rs = p$. Here λ is a formal symbol taken from some some index set Λ of size at most $|w|/\delta$. In order to avoid many case distinctions we consider the following (in some sense most interesting) special case, only. We assume that $\sigma(X)$ is a very long periodic word, $\sigma(Y)$ has a very long δ-periodic prefix, and $\sigma(Z)$ has a δ-periodic prefix longer than $|\sigma(X)|$, but no long δ-periodic suffix. Moreover, we assume that w has more than two very long δ-periodic factors. Note that $up^{e_\lambda} rv = urq^{e_\lambda} v$ if $1 \ne r \ne p$, $p = rs$, and $q = sr$. Let us resume: let $u_\lambda p_\lambda^{e_\lambda} r_\lambda v_\lambda$ be an occurrence of a very long δ-periodic factor in $\sigma(Z)$ with at least one visible position, $|u_\lambda| = |v_\lambda| = 3\delta$, and p_λ is primitive with $|p_\lambda| \le \delta$. Thus, $\lambda \in \Lambda$. There are three cases which we distinguish by using the names $\lambda, \nu, \rho \in \Lambda$. First, the occurrence of $u_\lambda p_\lambda^{e_\lambda} r_\lambda v_\lambda$ is the δ-periodic prefix of $\sigma(Z)$. As, by our simplification assumption, this prefix is longer than $\sigma(X)$, we deduce that we can write $\sigma(f, X) = u_\lambda p_\lambda^e p'$ with $p' \le p_\lambda$. Second, all "inner" positions $p_\nu^{e_\nu} r_\nu$ of $z = u_\nu p_\nu^{e_\nu} r_\nu v_\nu$ are visible. In this case, since $\sigma(X)$ (resp. $\sigma(Y)$) has a very long prefix (resp. suffix), this corresponds to an occurrence of the factor z in w. Third we can write $\sigma(g, Y) = p'' p_\rho^{e'} r_\rho v_\rho y$ with $e' \ge 6$ and $p'' \le p_\rho$ for some maximal δ-periodic factor $u_\rho p_\rho^{e_\rho} r_\rho v_\rho$ of $\sigma(Z)$ with $\rho \in \Lambda$. Moreover, we are in the case that the maximal δ-periodic prefix of $v_\rho y$ is v_ρ and $y \ne 1$. As we assumed that w has more than two very long δ-periodic factors, we can write $w = w_1 p_\nu^\nu r_\nu w_2$.

The procedure introduces at this point (for each $\lambda \in \Lambda$) new "typed" variables: $[X, p_\lambda]$, $[Y, \overline{p_\lambda}]$, and $[Z, p_\lambda]$. Actually, we need many more variables. Whenever we introduce variable $[V, p]$ we also introduce $[\overline{V}, \overline{p}]$ and $[V, qa]$ for $p = aq$; and we iterate this process. Finally, the action of G is defined by identifying $(f, [V, p])$ with $(1, [V, f(p)]) = [V, f(p)]$. (Note that $(f, [V, p]) = (g, [V, p]) \iff f(p) = g(p)$.)

The maximal number of these typed variables introduced by any equation with at most n variables is at most $2n|G|\delta$. The factor $2n$ is there because we consider for every variable a prefix and a suffix; and the factor δ comes in because $|p| \le \delta$ and with $p = aq$ every conjugate qa is also considered. Let $\mathcal{X}'$ be the enlarged set of untyped variables $X \in \mathcal{X}$ and fresh typed variables $[V, p]$. Together with introducing these variables we switch to the algebraic structure to read the equation in the monoid $M(B, \mathcal{X}', \theta, \mu')$ where the defining relations are given by $\theta = \{(a[V, p], [V, q]a \mid ap = qa \wedge a \in B\}$. We define the *type* of $[V, p]$ to be $\theta([V, p]) = p$ and we observe that the defining relations imply $p[V, p] = [V, p]p$ in $M(B, \mathcal{X}', \theta, \mu')$. We need a stronger notion of *solution* for typed variables in order to prevent that an unsolvable equation is transformed in a solvable one. If $\theta([V, p]) = p$, then we require $\sigma([V, p]) \in p^*$. Since σ is

a morphism, it also satisfies the defining relations. Hence, $a\sigma([V,p]) = \sigma([V,q])a$ implies $|\sigma([V,p])| = |\sigma([V,p])|$, too. The value of $\mu'([V,p])$ is defined implicitly in the following loop.

The loop is over all variables in some order. Of course, whatever happens to a variable V forces a simultaneous change in $\overline{V}$, too. We pop from each variable the maximal δ-periodic suffix of $\sigma(X)$ if this suffix is longer than 3δ. Otherwise we do nothing. As we have no control on the length of this suffix, we introduce a new typed variable. (Clearly, as we consider X and $\overline{X}$ prefixes and suffixes are popped out, and each X may produce two typed variable.) What we do in our concrete situation (where we have $\Lambda = \{\lambda, \nu, \rho\}$) is the following:

1. We substitute (f, X) by $\tau(f, X) = u_\lambda[X, p_\lambda]p_\lambda^\ell p'$. (So, X vanishes.) Moreover, for technical reasons, we require $5\delta < |p_\lambda^\ell p'| \le 6\delta$. We can define $\sigma'([X, p_\lambda]) \in p_\lambda^*$ such that $\sigma(f, X) = u_\lambda\sigma'([X, p_\lambda])p_\lambda^\ell p'$.
2. We substitute (g, Y) by $\tau(g, Y) = p''p_\rho^r[Y, p_\rho](g, Y)$ with the length condition $5\delta < \left|p''p_\rho^r\right| \le 6\delta$. We can define $\sigma'(g, Y) = v_\rho y$ and $\sigma'([Y, p_\rho]) \in p_\rho^*$ such that $\sigma(g, Y) = p''p_\rho^r\sigma'([Y, p_\rho])\sigma'(g, Y)$.
3. We substitute Z by $\tau(Z) = sq^{\ell'}[Z, q]Z$ with the length condition $5\delta < |sq^{\ell'}| \le 6\delta$. Here q is the conjugate of p_λ such that $q = sr_\lambda$, $p_\lambda = r_\lambda s$. We can define $\sigma'(Z) = v_\lambda z$ and $\sigma'([Z, q]) \in q^*$ with $sq^{\ell'}\sigma'([Z, q])\sigma'(Z) = \sigma(Z)$.

This leads to a new solution σ' to the twisted equation $u_\lambda[X, p_\lambda]p_\lambda^\ell p'wp''p_\rho^{r'}[Y, p_\rho](g, Y) = sq^{\ell'}[Z, q]Z$. We rename σ' as σ. Note that u_λ is a prefix of $sq^{\ell'}$. The positions of $[X, p_\lambda]$ and $[Z, q]$ are not adjusted, but our defining relations do not fix these positions. So, we use these defining relations to represent the equation by the following equation between words

$$u_\lambda[X, p_\lambda]p_\lambda^{\ell''}r_\lambda v_\lambda w'u_\rho p_\rho^{r'}r_\rho[Y, p_\rho](g, Y) = u_\lambda[Z, q]p_\lambda^{\ell'''}r_\lambda Z. \tag{3}$$

The morphism σ solves this equation. Moreover, in our concrete situation we have $v_\lambda w'u_\rho = vu_\nu p_\nu^\nu r_\nu u$; and again, we content ourselves to consider the special case where $u_\nu p_\nu^\nu r_\nu$ is the only occurrence of very long δ-periodic factor in $v_\lambda w'u_\rho$. Ignoring u_λ on the left, the remaining task is to compress the equation (where $v_\lambda \le vu_\nu$ and $\overline{u_\rho} \le \overline{u}\,\overline{v_\nu}$)

$$[X, p_\lambda]p_\lambda^{\ell''}r_\lambda vu_\nu p_\nu^\nu r_\nu v_\nu up_\rho^{r'}r_\rho[Y, p_\rho](g, Y) = [Z, q]p_\lambda^{\ell'''}r_\lambda Z \tag{4}$$

with respect to the solution σ. The crucial idea comes next: we use a larger alphabet of constants, we change the type of variables and we introduce more defining relations. For each $\lambda \in \Lambda$ we introduce a new constant, denoted as $[p_\lambda, r_\lambda, \lambda]$, and for each variable $[V, p]$ we introduce a constant $[p]$. Thus, $[p_\lambda, r_\lambda, \lambda]$ and $[p]$ are fresh letters. We also let act G on these letters in the obvious way, so we actually introduce more letters. Let h be the morphism defined by $h([p_\lambda, r_\lambda, \lambda]) = p_\lambda r_\lambda$ and $h([p]) = p$, it means h compresses the words $p_\lambda r_\lambda$ and p into single letters, then Equation (4) is the image under h of the equation

$$[X, p_\lambda][p_\lambda]^{\ell''-1}\,[p_\lambda, r_\lambda, \lambda]\,vu_\nu[p_\nu]^{\nu-1}\,[p_\nu, r_\nu, \nu]\,v_\nu u[p_\rho]^{r'-1}\,[p_\rho, r_\rho, \rho]\,[Y, p_\rho](g, Y) \tag{5}$$

$$= [Z, q][p_\lambda]^{\ell'''-1}\,[p_\lambda, r_\lambda, \lambda]\,Z \tag{6}$$

To have a visual notation we color the letters of the form $[p_\lambda, r_\lambda, \lambda]$ *green*. The procedure continues by redefining the type of a twisted variable $[V, p]$ as the letter $[p]$. We augment θ by more defining relations:

$$\{[V, p][p] = [p][V, p] \mid [V, p] \text{ twisted variable}\} \cup \left\{ [p_\lambda, r_\lambda, \lambda]\,[sr_\lambda] = [p_\lambda]\,[p_\lambda, r_\lambda, \lambda] \;\middle|\; p_\lambda = r_\lambda s \right\}.$$

It is not hard to see that we find a solution σ' of the new equation over the larger alphabet of constants such that $h\sigma' = \sigma h$ which is needed to prove the EDT0L property. The

remaining procedure is essentially the same as in [3]: using transformations either based on substitutions $[V,p] \mapsto [V,p][p]$ and $[V,p] \mapsto 1$ or homomorphisms based on $[p] \mapsto [p][p]$ and $[p_\lambda, r_\lambda, \lambda] \mapsto [p_\lambda, r_\lambda, \lambda]\,[p_\lambda]$ we can compress the above equation and simultaneously the solution such that the equation becomes its final form. We finish δ-periodic compression with

$$[p_\lambda, r_\lambda, \lambda]\, v u_\nu\, [p_\nu, r_\nu, \nu]\, v_\nu u\, [p_\rho, r_\rho, \rho]\, (g, Y) = [Z, q]\, [p_\lambda, r_\lambda, \lambda]\, Z. \tag{7}$$

The typed variables are gone, the letters $[p]$ are not visible anymore, moreover, the new solution doesn't use them. We are back in a free monoid, because none of the defining relations is used anymore. Note that Equation (7) is shorter than the original equation. Indeed, while the initial increase in the length of the equation is in $\mathcal{O}(n\delta)$, each green letter represents the inner part of a very long δ-periodic word of length at least 6δ.

▶ **Proposition 5.** *Let $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ be the state where we started δ-periodic-compression with $|W_s| \geq 8\delta n$; and let $E_t = (W_t, B_t, \mathcal{X}_t, \emptyset, \mu_t)$ the standard state where we finish δ-periodic-compression, and $(W, B, \mathcal{X}, \theta, \mu)$ any state which we have seen on the path from E_s to E_t during the procedure. Then we have $|W_t| \leq |W_s| + 20\delta n$ and $|W| \leq |W_s| + \mathcal{O}(\delta n)$. Moreover, let $n_{\text{new}} = \sum_{b \in B_t \setminus B_s} |W_t|_b$. If $n_{\text{new}} \geq 10n$, then $|W_t| < |W_s|$.*

▶ Remark. Note that n_{new} is the number of green letters we see in W_t. Let σ be the solution after δ-periodic-compression, then for $X \in \mathcal{X}_t$ the length of a δ-periodic prefix (and suffix resp.) is bounded by 3δ. Hence, there is no very long δ-periodic prefix or suffix in $\sigma(X)$.

3.4 Twisted pair-compression

We place ourselves after a sequence of rounds of popping out letters for each variable, alphabet-reduction, and δ-periodic compression. We are at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ where $\emptyset \neq \mathcal{X} \subseteq \mathcal{V}$. Without restriction, we may assume that $|W| \in \Theta(|G|n^2)$ and that the number of visible green letters is at most $10n$: our construction ensures that $|W| \in \mathcal{O}(|G|n^2)$, and we can always pop out letters to make the equation longer; and if the number of visible green letters exceeds $10n$ then according to Proposition 5 the most recent δ-periodic-compression had decreased the length of the equation, so we can perform another round.

Throughout, it is possible to write $W = U_1 \# u(f,X) w(g,Y) v \# U_2 \overline{U_2'} \,\overline{\#}\, \overline{v} \overline{Z} \overline{u} \,\overline{\#}\, \overline{U_1'}$ with $|U_i'|_\# = |U_i'|_{\overline{\#}}$, $i = 1, 2$. Here $u(f,X)w(g,Y)v = uZv$ is called a *local equation*. For simplicity we may assume that $u, v, w \in B^*$ and that $(f,X), (g,Y), Z = (1,Z)$ are twisted variables. Moreover, we may assume that for each local equation its "dual" equation $\overline{v}(g,\overline{Y})\overline{w}(f,\overline{X})\overline{u} = \overline{v}\overline{Z}\overline{v}$ is also part of the system encoded in W. Since W is long, we can assume that $|uwv|$ is long, too. Since there are at most $\mathcal{O}(n)$ green letters, there are long intervals without green letters. The goal is to compress enough pairs $ab \leq W$ of constants into single letters without causing any conflict or overlap with other pairs or variables that are connected via twisting. We compress pairs according to an equivalence relation between positions. The idea is that whenever we modify a solution at position i, then we must modify $\sigma(W)$ at all equivalent positions $j \equiv i$.

The notion of *equivalent positions* is defined for a given solution σ, it has a reasonably intuitive definition. We write $W = U\overline{V}$ with $\sigma(U) = \sigma(V)$ and $\sigma(U) = a_1 \cdots a_m$ with $a_i \in B$. We associate with U (resp. V) the interval $[1, m] \subseteq \mathbb{N}$ (resp. $[m+1, 2m]$) of positions and we let $i \sim m + i$ for $1 \leq i \leq m$. We say that position i sits directly "above" $m + i$, see Figure 2.

Each occurrence of a twisted variable (f, X) in UV corresponds to some interval of length $|\sigma(X)|$ in $[1, 2m]$ and we identify the i-th positions in each of these intervals for $1 \leq i \leq |\sigma(X)|$. Identified positions are represented by a unique position corresponding to

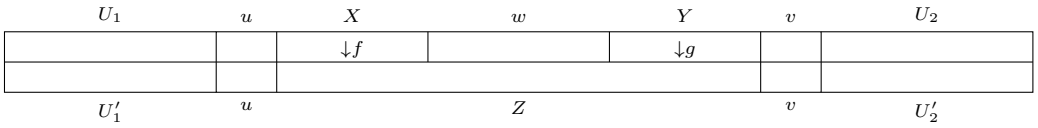


Figure 2 $W = U\overline{V}$ viewed as U on top and V on the bottom and $\sigma(U) = \sigma(V)$ in the middle.

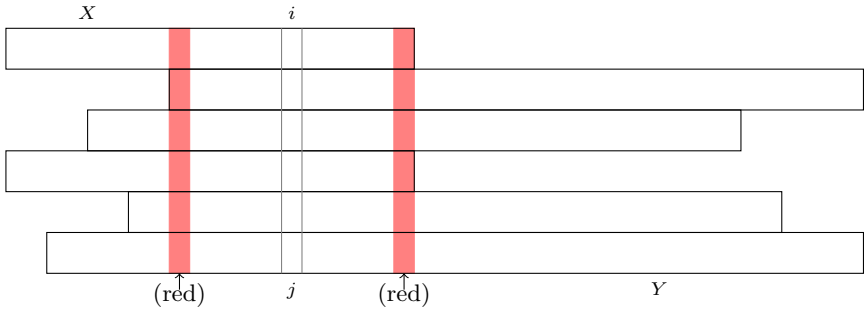


Figure 3 Red positions. We use $\sim$ to put $i \approx j$ into a "domino tower".

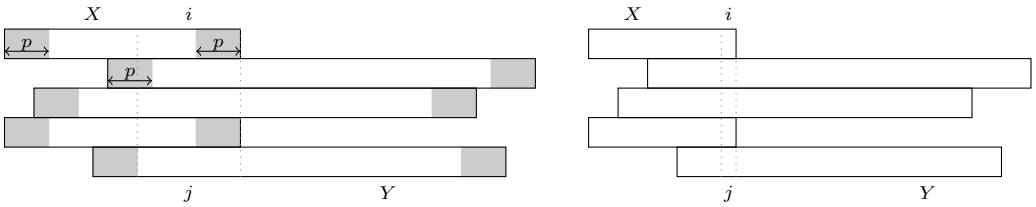
the leftmost occurrence of a twisted variable (f, X) in U. This interval is denoted by $I(X)$. Thus, we identify various positions and we carry over the relation $\sim$: if i and j are identified with i' and j' and if $i' \sim j'$, then we also let $i \sim j$. By $\approx$ we denote the generated equivalence relation of $\sim$. The relation $\approx$ can be visualized in so-called *domino towers* as in Figure 3. Clearly, we may have $i \approx j$ for various $i, j \in I(X)$. For example, an equation $(f, X)a = bX$ forces $i \approx j$ for all $i, j \in I(X)$. There is also a natural notion of duality: $I(X)$ and $I(\overline{X})$ are disjoint, but if we change $\sigma(X)$ at the first position, we must change $\sigma(\overline{X})$ at the last position. Thus, for the i-th position in $I(X)$ we let $\overline{i}$ be the $(|\sigma(X)| - i + 1)$-st position in $I(\overline{X})$; and we write $i \leftrightarrow \overline{i}$. Finally, we let $\equiv \subseteq [1, 2m] \times [1, 2m]$ be the equivalence relation generated by $\approx$ and $\leftrightarrow$. Clearly, if $i \approx j \leftrightarrow \overline{j}$ and the label at position i is $a \in B$, then a labels j and $\overline{a}$ labels $\overline{j}$.

Positions at the borders of some $\sigma(X)$ inside $\sigma(W)$ play a special role because we cannot compress over borders. We color the first and last position in each $I(X)$ *red* (unless it has already the color green) to signal "danger". We color red all positions equivalent to a red position, too. Since the set of green positions is closed under equivalence (they are the fresh letters $[p_\lambda, r_\lambda, \lambda]$), no conflict between red and green is introduced here. It follows that there are at most n pairwise different equivalence classes of red positions.

We extend the notion of equivalence to intervals (without red positions). Let $p \in \mathbb{N}$. We directly link an interval $[i, i + p]$ of positions in $\sigma(X)$ (resp. w, $\sigma(Y)$) to $[j, j + p]$ in $\sigma(Z)$ if there is an equation, for example like $u(f, X)w(g, Y)v = uZv$, such that $\sigma(X)[i, i + p]$ (resp. $w[i, i + p]$, $\sigma(Y)[i, i + p]$) sits directly above the $\sigma(Z)[j, j + p]$; and we write $[i, i + p] \sim [j, j + p]$ in this case. For each interval $[i, i + p]$ of positions in $\sigma(X)$ we also let $[i, i + p] \leftrightarrow [\overline{i + p}, \overline{i}]$. As above, we let $\approx$ and $\equiv$ be the generated equivalence relations of $\sim$ resp. $\sim \cup \leftrightarrow$. Since $i \sim j \iff \overline{i} \sim \overline{j}$ we can deduce

$$[i, i + p] \equiv [j, j + p] \iff [i, i + p] \approx [j, j + p] \vee [i, i + p] \approx [\overline{j + p}, \overline{j}]. \tag{8}$$

▶ **Lemma 6.** *Let $[i - 1, i, i + 1, i + 2]$ be an interval without any red position and where the four positions are pairwise inequivalent. Consider $[i, i + 1] \equiv [j, j + 1] \equiv [k, k + 1]$. Then either $k = j$ and $[j, j + 1] \not\approx [\overline{k} - 1, \overline{k}]$ or $[j, j + 1] \cap [k, k + 1] = \emptyset$.*



(a) $W : [i-p, i+p] \sim [j-p, j+p]$ **(b)** $W_p : i \sim j$

Figure 4 Example illustrating the proof of Lemma 7.

Let $p \in \mathbb{N}$ and σ be a solution for W. For each X we do:

- if $|\sigma(X)| \leq 2p$, then replace X by $\sigma(X)$ and remove X from the set of variables;
- if $|\sigma(X)| > 2p$, then write $\sigma(X) = uwv$ with $|u| = |v| = p$ and replace X by uXv. Change the interval $I(X) = [l, r]$ to $I_p(X) = [l + p, r - p]$. (So, it is smaller.)

Denote the new solution for W_p defined by that procedure by σ_p.

▶ **Lemma 7.** *Let i and j be positions in $\sigma_p(W_p) = \sigma(W)$ which belong to variables in W_p. This means $i, j \in \bigcup\{I_p(X) \mid X \in \mathcal{X}_p\}$. Then we have $i \sim j$ (resp. $i \leftrightarrow j$) for W_p and σ_p if and only if $[i-p, i+p] \sim [j-p, j+p]$ (resp. $[i-p, i+p] \leftrightarrow [j-p, j+p]$) for W and σ.*

We define and fix $\delta = |G|\,\varepsilon$ and $\varepsilon = 30n$. We start at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ together with a solution σ. For simplicity, we assume that all local equations have the form $u(f, X)w(g, Y)v = uZv$. Moreover, when we start pair-compression (directly after δ-periodic-compression) there are some green letters and corresponding green visible positions.

1. For every X in some order do: either replace X by $\sigma(X)$ (if $|\sigma(X)| \leq 10\delta$) or write $\sigma(X) = ux$ with $|u| = 10\delta$; replace X by $\tau(X) = uX$; rename the new equation and new solution as (E, σ). Define the intervals $I(X)$ as done above color *red* positions in $\sigma(W)$ which are equivalent to a first or last position in $I(X)$ unless they are green.
2. **while** there is an interval $[i-1, i, i+1, i+2]$ such that (1) all four positions are pairwise inequivalent, (2) no position is colored, and (3) all positions are visible **do**

   a. Let ab the label of the middle interval $[i, i+1]$. Choose fresh letter c and define a morphism h by $h(c) = ab$. (Hence, $f(c) = c \iff f(ab) = ab$, too.) Whenever $[i, i+1] \approx [j, j+1]$, then the label of $[j, j+1]$ is $f(ab)$ for some $f \in G$. Replace each of the intervals $[j, j+1]$ and $[\overline{j}-1, \overline{j}]$ by a single new position and label this position with $f(c)$ and $f(\overline{c})$ resp. There is no conflict in this relabeling by Lemma 6. Since there is no red position, there is no "crossing" of the intervals $[j, j+1]$ or $[\overline{j}-1, \overline{j}]$. So, this gives a new but shorter equation W'. We have $h(W') = W$ and new solution σ' such that $h\sigma'(W') = \sigma(W)$ There is a new numbering for the positions, but the colored positions can still be identified.

   b. Define $B' = B \cup \{f(c), f(\overline{c}) \mid f \in G\}$ and $E' = (W', B', \mathcal{X}, \emptyset, \mu')$.

   c. Rename (E', σ') as (E, σ) and transfer the induced coloring.

   end while

If we started the procedure with W and the while loop with W_ℓ, then the loop terminates with an equation W' and we introduced at most $|G|\,(|W_\ell| - |W'|)$ new letters. It is also clear that $|W'| \leq |W| + 20\delta n$ since any increase of length is due to the first steps, where we replaced each variable X either by $\sigma(X)$ or by uXv. The worst case for $|W'|$ is that no compression took place. However, we assume that there at most $10n$ green letters. Hence, we can use the following fact.

▶ **Proposition 8.** *Let (E, σ) with equation W just after a δ-periodic-compression where at most $10n$ green letters are visible. If $|W| \in 20\ell\delta n + \mathcal{O}(\delta n)$, then the pair-compression procedure outputs an equation W' such that $|W'| \leq |W| + 20\delta n$ and $|W'| \leq \frac{59|W|}{60} + \mathcal{O}(\delta n)$.*

Let us highlight that Proposition 8 is the key step in the proof of Theorem 1 and it is here where twisted conjugacy comes into play. Following any given solution at the initial state, it bounds the lengths of all intermediate equations in $\mathcal{O}(\delta n) = \mathcal{O}(|G|n^2)$. Since at a standard state we can perform an alphabet reduction we can bound the size of the extended alphabet C in $\mathcal{O}(|G|^2 n^2)$. Moreover, the number of untyped variables is never increasing. Typed variables disappear and reappear, but their number never grows beyond the size of C.

After δ-periodic compression, no $\sigma(X)$ started or ended in a very long δ-periodic word. In the procedure above either X vanished or we replaced X by uXv where $|u| = |v| = 10\delta$. We carefully colored some position red after that replacement. Consider the new equation with the new solution just after that step; and rename the corresponding pair as (W, σ). Consider positions $i < k$ in $\sigma(W)$ such that no position k with $i \leq k \leq j$ is green. With the help of Proposition 3, Lemma 7 and "domino towers" as depicted in Figure 4, one can show the following fact: if $i \equiv k \equiv j$ for some k, then $|j - i| > \varepsilon$. The fact is not obvious but extremely useful: knowing that equivalent positions are far apart allows one to find enough intervals of length four, such that pair-compression reduces their length to at most three by Lemma 6.

Putting all this together, the overall compression method has the following high-level description. Start at the initial state E_{init} with a given initial entire solution $(\text{id}_{A^*}, \sigma_{\text{init}})$.

begin compression
Rename E_{init} as $E = (W, B, \mathcal{X}, \emptyset, \mu)$; rename $(\text{id}_{A^*}, \sigma_{\text{init}})$ as (α, σ).
Repeat the following loop until $\mathcal{X} = \emptyset$.
begin loop
1. Pop out letters from variables until $|W| \geq 100\delta n$.
2. Define $\kappa > 0$ by $\kappa\delta n = |W|$. Call δ-periodic-compression (starting with an alphabet-reduction), and let W' denote the equation at the end of the procedure.
3. If $|W'| < \kappa\delta n$, then do nothing, else call pair-compression.
end loop
end compression

Proposition 8 implies that $\kappa \in \mathbb{Q}$ is bounded above by some effective constant in $\mathcal{O}(1)$. Defining a weight in $\mathbb{N}^4$ (ordered lexicographically) by

$$\|E, \alpha, \sigma\| = \left(\sum_{X \text{ has no type}} |\alpha\sigma(X)|, \sum_{X \text{ is typed}} |\alpha\sigma(X)|, |W|, |B| \right)$$

finally shows that the compression method terminates for every given solution because every step in the procedures is weight-reducing. This means our algorithm finds all solutions. This finishes the outline of the proof of Theorem 1.

Acknowledgements. We thank the anonymous referees for very helpful feedback.

References

1 Peter R. J. Asveld. Controlled iteration grammars and full hyper-AFL's. *Information and Control*, 34(3):248–269, 1977. doi:10.1016/S0019-9958(77)90308-4.

2 Gilbert Baumslag, Alexei Myasnikov, and Vladimir Remeslennikov. Algebraic geometry over groups. In *Algorithmic problems in groups and semigroups (Lincoln, NE, 1998)*, Trends Math., pages 35–50. Birkhäuser Boston, Boston, MA, 2000. doi:10.1007/978-1-4612-1388-8_3.

3 Laura Ciobanu, Volker Diekert, and Murray Elder. Solution sets for equations over free groups are EDT0L languages. *Internat. J. Algebra Comput.*, 26(5):843–886, 2016. Conference version in ICALP 2015, LNCS 9135. doi:10.1142/S0218196716500363.

4 François Dahmani and Vincent Guirardel. Foliations for solving equations in groups: free, virtually free, and hyperbolic groups. *J. Topol.*, 3(2):343–404, 2010. doi:10.1112/jtopol/jtq010.

5 Volker Diekert and Murray Elder. Solutions of twisted word equations, EDT0L languages, and context-free groups. *ArXiv e-prints*, January 2017. URL: https://arxiv.org/abs/1701.03297, arXiv:1701.03297.

6 Volker Diekert, Artur Jeż, and Wojciech Plandowski. Finding all solutions of equations in free groups and monoids with involution. *Inform. and Comput.*, 251:263–286, 2016. Conference version in Proc. CSR 2014, LNCS 8476 (2014). doi:10.1016/j.ic.2016.09.009.

7 Volker Diekert and Armin Weiß. Context-free groups and Bass-Serre theory. *ArXiv e-prints*, July 2013. arXiv:1307.8297.

8 Martin J. Dunwoody. The accessibility of finitely presented groups. *Inventiones Mathematicae*, 81(3):449–457, 1985. doi:10.1007/BF01388581.

9 Andrzej Ehrenfeucht and Grzegorz Rozenberg. On some context free languages that are not deterministic ET0L languages. *RAIRO Theor. Inform. Appl.*, 11:273–291, 1977.

10 Samuel Eilenberg. *Automata, languages, and machines. Vol. A*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York, 1974. Pure and Applied Mathematics, Vol. 58.

11 Julien Ferté, Nathalie Marin, and Géraud Sénizergues. Word-mappings of level 2. *Theory Comput. Syst.*, 54:111–148, 2014. doi:10.1007/s00224-013-9489-5.

12 Robert H. Gilman. Personal communication, 2012.

13 Robert H. Gilman, Susan Hermiller, Derek F. Holt, and Sarah Rees. A characterisation of virtually free groups. *Arch. Math. (Basel)*, 89(4):289–295, 2007. doi:10.1007/s00013-007-2206-3.

14 Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, volume 8 of *Math. Sci. Res. Inst. Publ.*, pages 75–263. Springer, New York, 1987. doi:10.1007/978-1-4613-9586-7_3.

15 Sanjay Jain, Alexei Miasnikov, and Frank Stephan. The complexity of verbal languages over groups. In *Proceedings of the 2012 27th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 405–414. IEEE Computer Soc., Los Alamitos, CA, 2012. doi:10.1109/LICS.2012.50.

16 Artur Jeż. Recompression: a simple and powerful technique for word equations. *J. ACM*, 63(1):Art. 4, 51, 2016. Conference version in Proc. STACS 2013. doi:10.1145/2743014.

17 Artur Jeż. Word Equations in Nondeterministic Linear Space, 2017. doi:10.4230/LIPIcs.ICALP.2017.95.

18 Abe Karrass, Alfred Pietrowski, and Donald Solitar. Finite and infinite cyclic extensions of free groups. *J. Austral. Math. Soc.*, 16:458–466, 1973. Collection of articles dedicated to the memory of Hanna Neumann, IV. doi:10.1017/S1446788700015445.

19 Dietrich Kuske and Markus Lohrey. Logical aspects of Cayley-graphs: the group case. *Ann. Pure Appl. Logic*, 131(1-3):263–286, 2005. doi:10.1016/j.apal.2004.06.002.

20 Markus Lohrey and Géraud Sénizergues. Theories of HNN-extensions and amalgamated products. In *Automata, languages and programming. Part II*, volume 4052 of *Lecture Notes in Comput. Sci.*, pages 504–515. Springer, Berlin, 2006. doi:10.1007/11787006_43.

21 Gennadiĭ S. Makanin. The problem of solvability of equations in a free semigroup. *Math. Sbornik*, 103:147–236, 1977. English transl. in Math. USSR Sbornik 32 (1977).

22 Victor Mazurov and Evgeny Khukhro. Unsolved Problems in Group Theory. The Kourovka Notebook. No. 18 (English version). *ArXiv e-prints*, January 2014. arXiv:1401.0300.

23 David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983. doi:10.1016/0022-0000(83)90003-X.

24 Wojciech Plandowski. Satisfiability of word equations with constants is in PSPACE. *Journal of the ACM*, 51:483–496, 2004. Conference version in FOCS'99. doi:doi:10.1145/990308.990312.

25 Wojciech Plandowski and Wojciech Rytter. Application of Lempel-Ziv encodings to the solution of word equations. In K. G. Larsen et al., editors, *Proc. 25th International Colloquium Automata, Languages and Programming (ICALP'98), Aalborg (Denmark), 1998*, volume 1443 of *Lecture Notes in Computer Science*, pages 731–742, Heidelberg, 1998. Springer-Verlag.

26 Eliyahu Rips and Zlil Sela. Canonical representatives and equations in hyperbolic groups. *Invent. Math.*, 120(3):489–512, 1995. doi:10.1007/BF01241140.

27 Grzegorz Rozenberg and Arto Salomaa. *The Book of L.* Springer, 1986.

28 Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of formal languages. Vol. 1.* Springer-Verlag, Berlin, 1997. Word, language, grammar. doi:10.1007/978-3-642-59126-6.

29 Géraud Sénizergues. An effective version of Stallings' theorem in the case of context-free groups. In *Automata, languages and programming (Lund, 1993)*, volume 700 of *Lecture Notes in Comput. Sci.*, pages 478–495. Springer, Berlin, 1993. doi:10.1007/3-540-56939-1_96.

30 Géraud Sénizergues. On the finite subgroups of a context-free group. In *Geometric and computational perspectives on infinite groups (Minneapolis, MN and New Brunswick, NJ, 1994)*, volume 25 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 201–212. Amer. Math. Soc., Providence, RI, 1996.

31 Jean-Pierre Serre. *Trees.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. Translated from the French original by John Stillwell, Corrected 2nd printing of the 1980 English translation.