SOLUTIONS TO TWISTED WORD EQUATIONS AND EQUATIONS IN VIRTUALLY FREE GROUPS

VOLKER DIEKERT AND MURRAY ELDER

Abstract. It is well known that the problem solving equations in virtually free groups can be reduced to the problem of solving twisted word equations with regular constraints over free monoids with involution. In this paper we prove that the set of all solutions of a twisted word equation is an EDT0L language whose specification can be computed in PSPACE. Within the same complexity bound we can decide whether the solution set is empty, finite, or infinite.

In the second part of the paper we apply the results for twisted equations to obtain in PSPACE an EDT0L description of the solution set of equations with rational constraints for finitely generated virtually free groups in standard normal forms with respect to a natural set of generators. If the rational constraints are given by a homomorphism into a fixed (or "small enough") finite monoid, then our algorithms can be implemented in $\mathsf{NSPACE}(n^2 \log n)$, that is, in quasi-quadratic nondeterministic space.

Our results generalize the work by Lohrey and Sénizergues (ICALP 2006) and Dahmani and Guirardel (J. of Topology 2010) with respect to both complexity and expressive power. Neither paper gave any concrete complexity bound and the results in these papers are stated for subsets of solutions only, whereas our results concern all solutions.

2010 Mathematics Subject Classification: 03D05, 20F65, 20F70, 68Q25, 68Q45.
Keywords: Equation in a virtually free group, twisted equation, EDT0L language, PSPACE.

Introduction

For a given semigroup S the decision problem WordEquation is the following: on input two words U and V in variables together with letters from a generating set $\Sigma \subseteq S$, decide whether or not there exists a substitution σ of variables by elements in S which yields a true identity $\sigma(U) = \sigma(V)$ in S. Here, σ is extended by $\sigma(s) = s$ for all $s \in \Sigma$.

In a seminal paper [36] Makanin showed that WordEquation is decidable for free semigroups. The first complexity estimation of the problem was a tower of several exponential functions, but this dropped down to PSPACE by Plandowski [43] using compression. The insight that long solutions of word equations can be efficiently compressed is due to [44] which also led to the still standing conjecture that WordEquation is NP-complete for free semigroups (and free groups). Until 2013 the known decidability proofs for solving word equations were long and technical with an accompanied reputation for being difficult. This changed drastically when Jeż applied his *recompression* technique: he presented an $\mathsf{NSPACE}(n \log n)$ algorithm to solve word equations [26][1]. Actually his method achieves more: it

Date: 1 December 2019.

Research supported by Australian Research Council (ARC) Project DP 160100486 and German Research Foundation (DFG) Project DI 435/7-1.

[1] In [27] Jeż improved the complexity to $\mathsf{NSPACE}(n)$.

describes all solutions, copes with rational constraints (which is essential in applications), it extends to free groups, and to free monoids with involution [11]. Refining Jeż's method, Ciobanu and the present authors showed that the full solution set of a given word equation over a free monoid with involution with rational constraints is EDT0L [5]. As a consequence, the same is true in free monoids without involution and for free groups where the constraints are used to ensure that solutions are given by reduced words. Previously this was only known for quadratic word equations [17]. EDT0L languages are defined by a certain type of Lindenmayer system. There is a vast literature on Lindenmayer systems, see [49], but all we need here is that an EDT0L language is specified by a nondeterministic finite automaton accepting endomorphisms over a free monoid and by some initial word. Applying the set of accepted endomorphisms to the initial word yields the language.

The original motivation for [5] was to prove that the full solution set in reduced words of equations in free groups is an indexed language, a problem which was open at that time [18, 25]. However, the result in [5] is stronger since EDT0L forms a strict subclass of indexed languages [15].

Transfer results as in [22, 5] from words to free groups have a long history. In the 1980s Makanin showed that the existential and positive theories of free groups are decidable [37]. In 1987 Razborov gave a description of all solutions for an equation in a free group via "Makanin-Razborov" diagrams [45, 46] which formed a cornerstone in the work of Kharlampovich/Myasnikov [30] and Sela [51] on the positive solution of Tarski's conjectures about the elementary theory in free groups.

The motivation for the present paper is along this line. We show that given a finitely generated virtually free group G there is an $\mathsf{NSPACE}(n^2 \log n)$ algorithm which produces for a given equation with (small) rational constraints an effective description of an EDT0L language which describes the solution set in standard normal forms over a natural set of generators. Moreover, the same complexity is enough to decide whether the solution set is empty, finite or infinite. No PSPACE algorithm, in fact no concrete complexity bound was known for deciding emptiness before.

In this paper, we define an $\mathsf{NSPACE}(s(n))$ algorithm to be a partially defined single-valued function f computed by a nondeterministic Turing machine consisting of three tapes: a one-way-read-only input tape, a two-way-read-write work tape, and a one-way-write-only output tape. If the length of the word written on the input tape is n, the work tape is restricted to having length $s(n)$. If the machine halts on input w at some point, then the contents w' on the output tape satisfies $w' = f(w)$. In general such a device might specify a partially defined *multi-valued* function, where several outputs are possible from the same input. However, in our case, we require that the output is unique. The domain of the partially defined function f computed by the machine is the halting set of the machine, and for each w in the domain there is a single output $f(w)$. This is the standard definition of a nondeterministic transducer which computes partially defined single-valued function. For nondeterministic polynomial time, formal definitions go back to [4]; see also [52, 53]. It is clear that this formalism applies to other nondeterministic complexity classes as well. Every $\mathsf{NSPACE}(s(n))$ transducer can be simulated by a deterministic transducer using at most working space $s(n)^2$ (Savitch's Theorem), and also by a deterministic Turing machine which uses a time bound in $2^{\mathcal{O}(s(n))}$, see [42] for more details. Thus, every PSPACE algorithm can be implemented such that it runs in deterministic singly exponential time $2^{\mathrm{poly}(n)}$.

Several remarks are in order here, which point to some additional difficulties in our framework. First, in general virtually free groups have torsion, which is a serious obstacle to applying the known techniques. The reason to study virtually free groups is motivated by the ubiquitous presence of word hyperbolic groups [20]. Solving equations in torsion-free hyperbolic groups reduces to solving equations in free groups [48], 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 [7]. The question of whether solving twisted word equations is decidable was asked by Makanin ([38] Problem 10.26(b)). It was solved in [7], thereby showing that the set of solvable equations over a f.g. virtually free group is decidable. This result was also independently shown by Lohrey and Sénizergues [35]. (Actually, [35] proves a more general transfer result.) What is in common: both papers are based on [10] and use explicitly ([35] only implicitly due to [10]) the so-called "exponent of periodicity". Because of this neither paper describes all solutions, nor gives any concrete complexity bounds.

The result for virtually free groups is obtained by a reduction to the problem to describe the solution sets of twisted word equations with regular constraints, following standard techniques. So, the main new contribution is our approach to solving twisted word equations. We follow the approach in [5] to define a sound and complete algorithm to produce an NFA $\mathcal{A}$ describing all solutions, however in the setting of twisted equations the technical details are quite far from previous methods. For example, for readers familiar with previous methods, in twisted equations it does not make sense to "uncross" pairs ab where a, b are different letters because once all pairs ab are uncrossed the twisting may produce new crossing pairs ba, uncrossing them leads to new crossing pairs ab etc. Thus, our underlying method is quite different from the original recompression due to Jeż.

The class of f.g. virtually free groups appears in many different ways. For example, a fundamental theorem of Muller and Schupp (relying originally on [14]) says that a f.g. group is virtually free if and only if it is *context-free* [40]. 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 [29], (2) f.g. groups having a Cayley graph with finite treewidth [32], (3) universal groups of finite pregroups [47], (4) groups having a finite presentation by some geodesic string rewriting system [19], and (5) f.g. groups having a Cayley graph with decidable monadic second-order theory [32]. Proofs for the most important equivalences are in [13]. The transformations are effective. For example, starting from a context-free grammar for the word problem, we can construct a representation as a fundamental group of finite graphs of finite groups. However, the finite graphs of finite groups can be much larger than the size of the context-free grammar: the result in [54] showed a primitive recursive bound on the blow-up. It was only very recently that Sénizergues and Weiß showed in [56] that the blow-up can be bounded by a doubly exponential function.

What we use here is another characterization which is proved in [13, Sec. 2.4.5]. It follows rather easily from Bass-Serre theory [57] and the representation of a f.g. virtually free group as a fundamental group of finite graphs of finite groups. The characterization says that a f.g. group G is virtually free if and only if it has an effective embedding into a semi-direct product of a free group F with basis E_+ by a finite group H which acts by permutations on the symmetric set $E = E_+ \cup E_+^{-1}$. (The precise statement is in Proposition 14.7.) Taking this characterization as a black box, no knowledge in Bass-Serre theory is required to understand our results.

An extended abstract of a preliminary version of this paper was presented at the conference ICALP 2017, Warsaw (Poland), 10-14 July 2017 [9]. Ciobanu and the second author have now extended the results of the present paper to show solutions to equations in any hyperbolic group are EDT0L with description in PSPACE [6].

1. Organization of the paper

1.1. **The overall structure.** The paper has two main and separate parts. In a first part we deal the following algorithmic problem. The input is a system $\mathcal{S}$ of twisted word equations with regular constraints over a free monoid (with involution) A^*. The "twist" comes from a finite group H acting on A^*. We present a PSPACE algorithm which constructs an NFA $\mathcal{A}_\mathcal{S}$ which gives a description of the set of all solutions as an EDT0L language. Structural properties of the NFA $\mathcal{A}_\mathcal{S}$ tell us whether the set of all solutions is empty, finite, or infinite. Precise complexity bounds are discussed in Section 4.2, and under certain assumptions on the size of the regular constraints, we prove the entire algorithm can be done in $\mathsf{NSPACE}(|H|n^2 \log |A| \log n)$ where n denotes the input size of $\mathcal{S}$, and when $|A|$ and $|H|$ are constants, this becomes quasi-quadratic nondeterministic space $\mathsf{NSPACE}(n^2 \log n)$.

In a second part we apply our results on twisted word equations to the existential theory of equations with rational constraints over finitely generated virtually free groups. From the algorithmic viewpoint we deal with a non-uniform complexity where the virtually free group G is not part of the input. (This allows us to assume that G is embedded into a semi-direct product of a free group F by a finite group H, where the rank of F and the size of H are constants.) The input is a Boolean formula Φ in free variables over equations and rational constraints for the solution specified by NFAs which accepts subsets of G. The output is a specification of the set of all solutions in standard normal forms for Φ as an EDT0L language. The proof is a reduction to the setting in the first part. The result is in the same overall complexity as in the first part when taking into account that H, F and A are not part of the input.

In the final section we perform this reduction explicitly for the special linear group $\mathrm{SL}(2, \mathbb{Z})$ (without relying on any knowledge of Bass-Serre theory) starting with the well-known classical fact that $\mathrm{SL}(2, \mathbb{Z})$ can be embedded in semi-direct product of a free group of rank 2 (its commutator subgroup) by the finite cyclic group $\mathbb{Z}/12\mathbb{Z}$. *A priori*, there could be an exponential blow-up in the complexity due to fact that we use matrices and not a word representation when describing equations over $\mathrm{SL}(2, \mathbb{Z})$. However, there is no such blow-up thanks to work of Gurevich and Schupp [21].

1.2. **Technical details.** We assume that the reader is familiar with some basic facts in combinatorics on words, formal languages and finite automata, and complexity theory. Apart from that (and the promise that a finitely generated (f.g. for short) virtually free group admits an embedding into a certain semi-direct product of a free group F by a finite group H) the paper is self-contained. In principle, it is not necessary that the reader has ever heard of Makanin, word equations, or any method to solve them before. The paper uses various technical tools where the authors would have preferred to give references in the literature rather than lengthy and somewhat pedestrian constructions, but failed to find the appropriate references.

The heart of the paper is Jeż's compression method in the framework of twisted equations: Section 10 and Section 11. The adaption to the twisted setting is far from trivial and quite different from the original method in [26] or its extension to free groups as in [11] or [5]. Therefore to understand Sections 10 and 11 is the most demanding part when reading the paper.

Many of the technicalities surrounding NSPACE complexity can be overlooked if the reader is happy enough to replace the explicit complexity bounds by PSPACE.

2. Preliminaries

We use standard notation. If A and B are sets, then $A \subseteq B$ means set inclusion, while $A \subsetneq B$ means $A \subseteq B \wedge A \neq B$. By $A \setminus B$ we denote the set of $a \in A$ which are not in B. By B^A we mean the set of mappings from A to B, and 2^A denotes the power set of A, that is, $2^A = \{B \mid B \subseteq A\}$. We also view 2^A as a commutative and idempotent monoid.

$\mathbb{B} = (\{0,1\}, \max, \cdot, 0, 1)$, denotes the Boolean semi-ring, $\mathbb{N}$ (resp. $\mathbb{Z}$) denotes the semi-ring of natural numbers, (resp. the ring of integers). $\mathbb{N}$ is also the free monoid and $\mathbb{Z}$ is the free group in one generator.

Monoids (resp. groups) will typically be denoted by M and N (resp. by G and H). If the focus is on finite monoids (resp. finite groups), then we use the notation N (resp. H). With a few exceptions (like $\mathbb{N}$ or $\mathbb{Z}$) we denote the identity element in monoids by 1. A *zero* in a monoid M is an element $0 \in M$ such that $0x = x0 = 0$ for all $x \in M$. If a zero exists, it is unique. Nontrivial groups cannot have a zero.

Let M be a monoid and $u, v \in M$. We say that u is a *factor* of v if we can write $v = xuy$ for some $x, y \in M$. If we can write $v = uy$ (resp. $v = xu$), then we say that u is a *prefix* (resp. *suffix*) of v. If u is a prefix of v, then we also write $u \leqslant v$.

2.1. Complexity.

The $\mathcal{O}$-notation and complexity classes like P, NP, $\mathsf{NSPACE}(s(n))$, PSPACE are defined as in standard textbooks ([24, 42]), see also page 2. We use a convention that $\log(m) = \max\{1, \log_2(m)\}$. Throughout we use the well-known fact due to Immerman-Szelepcsényi that $\mathsf{NSPACE}(s(n))$ is closed under complementation. Note that any statement about complexity depends on how the input is given. A statement like "factorization is not known to be in P" makes sense only if the encoding of the problem and a notion of *input size* has been defined. If integers are encoded in unary, then, trivially, "factorization is in P" is true. Typically, inputs have various parameters. If certain parameters of the input are fixed, then with respect to the input size these parameters behave as constants. Still, for many problems $\mathcal{P}$ it is more accurate to use parametrized inputs, where the input size is tuple of non-negative numbers: $\|\mathcal{P}\| = (p_1, \ldots, p_k)$ with $k \geqslant 1$. If $\mathcal{P}$ is such a problem, then with respect to polynomial resource bounds we view $\mathcal{P}$ as a one-parameter problem of input size $n = 1 + p_1 + \cdots + p_k$. The notation is robust: every polynomial in $(1+p_1)^\ell \cdots (1+p_k)^\ell$ is also a polynomial in n for all $\ell \geqslant 1$. Throughout, we take care to define our input sizes (of systems of equations, or Boolean formulae, with regular constraints) in a natural way.

2.2. Sets and monoids with involution.

An *involution* of a set is a bijection $x \mapsto \overline{x}$ such that $\overline{\overline{x}} = x$ for all x in the set. The identity map is an involution. A *monoid with involution* additionally has to satisfy $\overline{xy} = \overline{y}\,\overline{x}$. This implies $\overline{1} = 1$ and $\overline{0} = 0$ (in case there is a zero). If G is a group, then it is a monoid with involution by taking $\overline{g} = g^{-1}$ for all $g \in G$. By default, we choose $\overline{g}$ to be g^{-1} in groups.

A *morphism* between sets with involution is a mapping respecting the involution. A *morphism* between monoids with involution is a homomorphism $\varphi\colon M \to M'$ such that $\varphi(\overline{x}) = \overline{\varphi(x)}$. Note that every group homomorphism is a morphism of monoids with involution. The set of automorphisms on a set (or monoid) M forms the group $\mathrm{Aut}(M)$. For $\Delta \subseteq M \cap M'$ we say that $\varphi\colon M \to M'$ is a Δ-*morphism* if $\varphi(x) = x$ for all $x \in \Delta$.

2.3. Group actions and H-monoids.

Recall that a group H acts on a set Σ (with involution) via a homomorphism $\psi\colon H \to \mathrm{Aut}(\Sigma)$. That is, ψ defines a permutation $x \mapsto g{\cdot}x$ with $1{\cdot}x = x$, $f{\cdot}(g{\cdot}x) = (fg){\cdot}x$ (and $f{\cdot}\overline{x} = \overline{f \cdot x}$) for all $f, g \in H$ and $x \in \Sigma$. Thus, every $g \in H$ defines a permutation of Σ (which respects the involution). The *stabilizer* of $x \in \Sigma$ is the subgroup $H_x = \{g \in H \mid g \cdot x = x\}$. Frequently, we also write $g(x)$ as a synonym for $g \cdot x$. If H acts on a monoid M, then we additionally demand that every element of H acts as an automorphism: $g(xy) = g(x)g(y)$. If M is equipped with an involution, then we have $g(\overline{xy}) = g(\overline{y})g(\overline{x}) = \overline{g(y)}\;\overline{g(x)}$. In the following we say that M is an *H-monoid* if it is a monoid with involution on which H acts. A morphism between H-monoids M and M' is given by an *H-compatible morphism* which is a homomorphism $\varphi\colon M \to M'$ respecting for all $g \in H$ and $x \in M$ the action $g \cdot \varphi(x) = \varphi(g \cdot x)$, and the involution, $\varphi(\overline{x}) = \overline{\varphi(x)}$.

2.4. **Free monoids with involution and an H-action.** By an *alphabet* we mean a finite set Σ with involution. (Since the identity is an involution of Σ this covers the case of monoids without a predefined involution.) The elements of Σ are called *letters* (or *symbols*). By Σ^* (by Σ^+ resp.) we denote the free monoid (free semigroup resp.) over Σ. The elements of a free monoid are called *words.* The empty word in a free monoid is also denoted by 1 as in other monoids. We have $\Sigma^+ = \Sigma^* \setminus \{1\}$. The involution extends to Σ^*: for a word $w = a_1 \cdots a_m$ with $a_i \in \Sigma$ we let $\overline{w} = \overline{a_m} \cdots \overline{a_1}$. The monoid Σ^* is called the *free monoid with involution over* Σ. If $\overline{a} = a$ for all $a \in \Sigma$ then $\overline{w}$ is simply the word w read from right-to-left. The length of word w is denoted by $|w|$, and $|w|_a$ counts how often a letter a appears in w.

If a group H acts on Σ, then $g \in H$ acts on $w = a_1 \cdots a_m$ with $a_i \in \Sigma$ by

$$g(w) = g(a_1) \cdots g(a_m).$$

A letter $a \in \Sigma$ is *H-visible* in w if $g(a)$ is a factor of w for some $g \in H$.

Sometimes it is useful to view a word of $w = a_1 \cdots a_m$ with $a_i \in \Sigma$ as a labeled linear order as follows. We let $\{1, \ldots, m\}$ be set of *positions*; and we label a position $1 \leqslant p \leqslant m$ with the letter $w[p] = a_p$. For $1 \leqslant i, j \leqslant m$ we denote by $[i, j]$ the *interval* $\{i, \ldots, j\}$. The labels of the interval define a factor

$$w[i, j] = a_i \cdots a_j.$$

An *occurrence* of a factor u in w is an interval $[i, j]$ such that $u = w[i, j]$. Typically, a factor has several occurrences. For a position $1 \leqslant p \leqslant |w|$ we define its *dual* position $\overline{p}$ by $\overline{p} = m + 1 - p$. The notion of duality extends to intervals $[i, j]$ with $1 \leqslant i, j \leqslant |w|$ by $\overline{[i, j]} = [\overline{j}, \overline{i}]$. Thus, the set of intervals is a set with involution.

A word w such that $w = \overline{w}$ is called *self-involuting*, and for such w we have $\overline{w[i, j]} = w[\overline{j}, \overline{i}]$.

2.5. **Automata, rational and recognizable subsets in a monoid.** For notation and results in the this subsection we refer to the classical textbook [16]. A *regular language* in finitely generated free monoids can be defined via a *nondeterministic finite automaton* or via recognizability using a homomorphism to a finite monoid, to mention just two possible definitions. We need the corresponding notions subsets for other monoids, too.

Let M be any monoid (not necessarily equipped with an involution). A *nondeterministic automaton* over M is a directed arc-labeled graph $\mathcal{A}$ denoted as a tuple $\mathcal{A} = (Q, M, \delta, \mathscr{I}, \mathscr{F})$. The vertices of $\mathcal{A}$ form the set Q of *states*, with subsets $\mathscr{I}$ of *initial* and $\mathscr{F}$ *final* states. We write $\mathcal{A} = \emptyset$ if there are no states. The arcs are called *transitions* and they are labeled with elements of the monoid M. We represent the set of transitions δ as a subset of $Q \times M \times Q$. A transition labeled by $1 \in M$ is called an *ε-transition.* In pictures we draw a transition (p, h, q) as $p \xrightarrow{h} q$. 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 labels together yields m. This defines the *accepted language* $L(\mathcal{A}) = \{m \in M \mid m \text{ is accepted by } \mathcal{A}\}$.

Often we specify M together with a set Σ of generators or, more generally, together with homomorphism π from the free monoid Σ^* to M. In that case, we may denote $\mathcal{A}$ alternatively as $\mathcal{A} = (Q, \Sigma, \delta, \mathscr{I}, \mathscr{F})$ where $\delta \subseteq Q \times \Sigma^* \times Q$. This allows two natural interpretations of $L(\mathcal{A})$: first as the set of words $L(\mathcal{A}) \subseteq \Sigma^*$ obtained by reading $\mathcal{A}$ as a shorthand for $(Q, \Sigma^*, \delta, \mathscr{I}, \mathscr{F})$; second as $L(\mathcal{A}) \subseteq M$ by identifying $L(\mathcal{A}) \subseteq \Sigma^*$ with $\pi(L(\mathcal{A}))$. If the distinction is crucial, we write $L(\mathcal{A})$ and $\pi(L(\mathcal{A}))$. However, sometimes a sloppy notation $L(\mathcal{A}) \subseteq M$ is used. There will be however no risk of confusion.

A *subautomaton* $\mathcal{A}'$ of $\mathcal{A} = (Q, M, \mathscr{I}, \mathscr{F})$ is an automaton $\mathcal{A}' = (Q', M, \mathscr{I}', \mathscr{F}')$ such that $Q' \subseteq Q$, $\delta' \subseteq \delta$, $\mathscr{I}' \subseteq \mathscr{I}$, and $\mathscr{F}' \subseteq \mathscr{F}$.

An automaton is called *trim* if every state is on some path from an initial to a final state. For a trim automaton $\mathcal{A}$ we have $L(\mathcal{A}) \neq \emptyset$ if and only if $\mathcal{A} \neq \emptyset$. Clearly, every automaton $\mathcal{A}$ contains a trim subautomaton $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A})$.

If the set of transitions is finite, then we call $\mathcal{A}$ a *nondeterministic finite automaton* (or *NFA* for short). A subset $L \subseteq M$ is called *rational* if L is accepted by some NFA over M.

A subset $L \subseteq M$ is called *recognizable* if there is a homomorphism $h \colon M \to N$ to a finite monoid N such that $h^{-1}(h(L)) = L$. In case that M is a finitely generated free monoid, the notion of rational and recognizable subsets coincide; so these subsets are also called *regular.* It follows rather easily that a monoid M is finitely generated if and only if all recognizable subsets are rational [39]. Finite subsets are always rational, but finite subsets in a group are recognizable if and only if the group is finite.

2.6. **From NFAs to Boolean matrices.** Nondeterministic finite automata encode regular languages in a concise and natural way. It is convenient to work in an algebraic framework with recognizing morphisms, too. Let us recall a well-known and classical construction.

Let $\mathcal{A} = (Q, \Sigma, \delta, \mathscr{I}, \mathscr{F})$ be any NFA with m states. Then we can assume that $Q = \{1, \ldots, m\}$, and we represent transitions as a mapping to Boolean $m \times m$ matrices as follows. For each letter $a \in \Sigma$ we define a matrix $\mu_{\mathcal{A}}(a) \in \mathbb{B}^{m \times m}$ by

$$(\mu_{\mathcal{A}}(a))_{s,t} = 1 \iff a \in L(Q, \Sigma, \delta, \{s\}, \{t\}). \tag{1}$$

We obtain a homomorphism $\mu_{\mathcal{A}} \colon \Sigma^* \to \mathbb{B}^{m \times m}$ such that for all $w \in \Sigma^*$ we have

$$w \in L(\mathcal{A}) \iff \mu_{\mathcal{A}}(w) \in \mu_{\mathcal{A}}(L(\mathcal{A}) \iff \exists s \in \mathscr{I}\ \exists t \in \mathscr{F} : (\mu_{\mathcal{A}}(w))_{s,t} = 1).$$

Example 2.1. *Let Σ be an alphabet (with involution) and $H \leqslant \mathrm{Aut}(\Sigma)$ be a subgroup of automorphisms. The set R of words having a factor $e\overline{e}$ for some $e \in \Sigma$ is regular and R is invariant under the action of H. Let $\mathbb{F} = \Sigma^* \setminus R$ be the complement: it is the set of* reduced *words. The set $\mathbb{F}$ is in canonical bijection with the group $\Sigma^* / \{e\overline{e} = 1 \mid e \in \Sigma\}$. The language R is accepted by an NFA (actually a DFA) with $1 + |\Sigma|$ states. Hence, $\mathbb{B}^{m \times m}$ recognizes them where $m = 1 + |\Sigma|$.*

The size of $\mathbb{B}^{m \times m}$ is 2^{m^2}, but there is a much smaller monoid N which recognizes $R = \bigcup \{\Sigma^* e\overline{e}\Sigma^* \mid e \in \Sigma\}$ and hence $\mathbb{F}$, too. The elements of N are 1, 0, and the pairs (a,b) in $\Sigma \times \Sigma$. The elements 1 and 0 act as the neutral element and a zero, respectively. The multiplication for the other elements is given by $(a,b) \cdot (c,d) = (a,d)$ if $b \neq \overline{c}$ and $(a,b) \cdot (c,d) = 0$, otherwise. The involution is given by $\overline{(a,b)} = (\overline{b}, \overline{a})$. In effect, N "remembers" the first and last letters of elements in $\mathbb{F}$. A pair (a,b) switches to 0 once a factor $e\overline{e}$ is recorded.

It is an H-monoid by the natural action of H induced by the action of H on Σ. Consider the morphism of H-monoids $\mu \colon \Sigma^* \to N$ which is defined by $\mu(a) = (a,a)$. Then we have $R = \mu^{-1}(0)$ and $\mathbb{F} = \mu^{-1}(N \setminus \{0\})$. The size of N is therefore $2 + |\Sigma|^2 - |\Sigma|$. Therefore each element in N can be specified by at most $1 + \log_2(1 + |\Sigma|)$ bits.

3. Regular languages in presence of an involution and an H-action

The application of this section is stated precisely in Proposition 3.2. We give a construction which allows us to handle regular constraints for twisted word equations using morphisms to finite H-monoids.

We proceed in two steps. The first step forces homomorphisms to respect the involution. This part is from the arXiv version of [5] which was inspired by [10]. We repeat that construction. Let N be any monoid. We define its dual monoid N^{op} to use the same set $N^{\mathrm{op}} = N$, but N^{op} is equipped with a new multiplication $x \circ y = yx$. In order to indicate whether we view an element in the monoid N or N^{op}, we use a flag: for $x \in N$ we write x^{op} to indicate the same element in N^{op}. Thus, we can suppress the symbol $\circ$ and we simply write $x^{\mathrm{op}} y^{\mathrm{op}} = (yx)^{\mathrm{op}}$. The notation is intended to mimic transposition in matrix calculus, where the dual operation is just the transpose. Similarly, we write 1 instead of 1^{op} which is true for the identity matrix as well. The direct product $N \times N^{\mathrm{op}}$ becomes a monoid with involution by letting $\overline{(x, y^{\mathrm{op}})} = (y, x^{\mathrm{op}})$. Indeed,

$$\overline{(x_1, y_1^{\mathrm{op}}) \cdot (x_2, y_2^{\mathrm{op}})} = (y_2 y_1, (x_1 x_2)^{\mathrm{op}}) = \overline{(x_2, y_2^{\mathrm{op}})} \cdot \overline{(x_1, y_1^{\mathrm{op}})}.$$

The following observations are immediate.

- If N is finite then $N \times N^{\mathrm{op}}$ is finite, too.
- We can embed N into $N \times N^{\mathrm{op}}$ by a homomorphism $\iota\colon N \to N \times N^{\mathrm{op}}$ defined by $\iota(x) = (x, 1)$. Note that if $\eta\colon N \times N^{\mathrm{op}} \to N$ denotes the projection onto the first component, then $\eta\iota = \mathrm{id}_N$.
- If M is a monoid with involution and $\rho\colon M \to N$ is a homomorphism of monoids, then we can lift ρ uniquely to a morphism $\varphi^{\mathrm{op}}\colon M \to N \times N^{\mathrm{op}}$ of monoids with involution such that we have $\rho = \eta\varphi^{\mathrm{op}}$. Indeed, it is sufficient and necessary to define $\varphi^{\mathrm{op}}(x) = (\rho(x), \rho(\overline{x})^{\mathrm{op}})$.

Example 3.1 ([10])**.** *Let $M = \mathbb{B}^{n\times n}$. Then $M \times M^{\mathrm{op}}$ is a submonoid of the set of $2n \times 2n$-Boolean matrices:*

$$\mathbb{B}^{n\times n} \times (\mathbb{B}^{n\times n})^{\mathrm{op}} = \left\{ \left(\begin{smallmatrix} P & 0 \\ 0 & Q^T \end{smallmatrix}\right) \;\middle|\; P, Q \in \mathbb{B}^{n\times n} \right\} \text{ with } \overline{\left(\begin{smallmatrix} P & 0 \\ 0 & Q^T \end{smallmatrix}\right)} = \left(\begin{smallmatrix} Q & 0 \\ 0 & P^T \end{smallmatrix}\right).$$

In the line above P^T and Q^T are the transposed matrices.

Now, we switch to the new part of our construction. For readers familiar with wreath products it might be helpful to say that the following is a wreath product construction. Let N be a monoid with involution. Consider the direct product N^H, which is the set of maps from H to N. We denote the elements of N^H by tuples $(n_g)_g$ with the interpretation that $g \in H$ is mapped to $n_g \in N$. It is a monoid by pointwise multiplication with involution $\overline{(n_g)_g} = (\overline{n_g})_g$. The monoid N embeds into N^H by sending n to the constant map $(n)_g$. We let act H on N^H by

$$f \cdot (n_g)_g = (n_{gf})_g.$$

Now, let M be an H-monoid with involution and let $\psi\colon M \to N$ be a morphism of monoids with involution, then we extend it to $\widetilde{\psi}\colon M \to N^H$ by

$$\widetilde{\psi}(x) = (\psi(gx))_g.$$

The homomorphism $\widetilde{\psi}$ respects the involution since

$$\widetilde{\psi}(\overline{x}) = (\psi(g\overline{x}))_g = \overline{(\psi(gx))_g};$$

and it respects the action of H since

$$\widetilde{\psi}(fx) = (\psi(gfx))_g = f \cdot \widetilde{\psi}(x).$$

$$\begin{tikzcd}
& (N \times N^{\mathrm{op}})^H \arrow[d, "\eta_1"] \\
& N \times N^{\mathrm{op}} \arrow[d, "\eta"] \\
M \arrow[ruu, "\widetilde{\varphi}"] \arrow[ru, "\varphi^{\mathrm{op}}"] \arrow[r, "\varphi"] & N
\end{tikzcd}$$

FIGURE 1. A lifting of a homomorphism φ to an H-compatible morphism $\widetilde{\varphi}$.

Moreover, ψ factorizes though $\widetilde{\psi}$ because $\widetilde{\psi}(x) = (\psi(gx))_g$ implies $\psi = \eta_1\widetilde{\psi}$ where $\eta_1((n_g)_g) = n_1$.

If we start with a homomorphism φ from an H-monoid with involution M to a monoid N without involution, then ψ means the morphism $\varphi^{\mathrm{op}}\colon M \to N \times N^{\mathrm{op}}$; and $\widetilde{\varphi}$ is a shorthand for $\widetilde{\varphi^{\mathrm{op}}}$. Thus, the constructions above yield the commutative diagram as in Figure 1. In that figure $\varphi = \eta\eta_1\widetilde{\varphi}$ is a homomorphism, $\varphi^{\mathrm{op}} = \eta_1\widetilde{\varphi}$ is a morphism of monoids with involution, and $\widetilde{\varphi}$ is an H-compatible morphism. As a direct consequence we obtain the following proposition. Recall that an H-monoid is, by definition, a monoid with involution.

Proposition 3.2. *Let H be a finite group that acts on a finite alphabet A; and hence via a length and involution preserving action on A^*. Then all recognizable subsets of A^* can be recognized by some H-compatible morphism to a finite H-monoid. More precisely, let $\varphi\colon A^* \to N$ be a homomorphism to a finite monoid; and $L = \varphi^{-1}(F)$ for some $F \subseteq N$. Then we have $L = \widetilde{\varphi}^{-1}(\widetilde{F})$ where $\widetilde{F} = \widetilde{\eta}^{-1}(F)$ and $\widetilde{\eta} = \eta\eta_1$.*

3.1. **Stabilizers.** Let H be a finite group acting on an alphabet A via a homomorphism $\psi\colon H \to \mathrm{Aut}(A)$. We assume that H is given by its multiplication table. The table can be stored with $\mathcal{O}(|H|^2 \log |H|)$ bits. We also need a way to represent the action of H and the stabilizer subgroups of H. The action is recorded by writing down for each $f \in H$ the element $\psi(f)$ as a permutation of A. To do this, we write $\psi(f)$ as a set of pairs $\psi(f) = \{(a, f(a)) \mid a \in A\}$. Thus, the action of H on A can be stored with $\mathcal{O}(|H|\,|A| \log |A|)$ bits.

For a word $w \in A^*$ we denote by H_w its *stabilizer*:

$$H_w = \{f \in H \mid f(w) = w\}\,.$$

stabilizer are subgroups; and the set of subgroups of the form H_w form a commutative monoid $\mathrm{ST}(H)$ where the operation is intersection, the identity element is H, and the involution is the identity. Indeed, we have

$$H_{uv} = \{g \in H \mid g(uv) = g(u)g(v) = uv\} = H_u \cap H_v,$$

$$H_{\overline{u}} = \{g \in H \mid g(\overline{u}) = \overline{g(u)} = \overline{u}\} = H_u$$

and

$$H_{g(u)} = \{f \in H \mid f(g(u)) = g(u)\} = \{f \in H \mid g^{-1}(f(g(u))) = u\} = g^{-1}H_u g$$

for all $u, v \in A^*$, $g \in H$. In particular, H acts on $\mathrm{ST}(H)$ by conjugation, and $\mathrm{ST}(H)$ is therefore an H-monoid. Let $\mathrm{SG}(H)$ denote the set of all subgroups of H, then $\nu(u) = H_u$ yields a canonical surjective morphism

$$\nu\colon A^* \to \mathrm{ST}(H) = \{H_w \in \mathrm{SG}(H) \mid w \in A^*\}\,.$$

A basic test is to answer "$f \in H_w$?". This is easy: for $w = a_1 \cdots a_m$ with $a_i \in A$ we check one after another that $f(a_i) = a_i$ for $1 \leqslant i \leqslant |w|$. This enables an efficient test to decide whether or not $H_u \subseteq H_v$. For each $f \in H$ one after another we test whether $f \in H_u$ implies $f \in H_v$. In particular, we can answer "$H_u = H_v$?".

Lemma 3.3. *We have* $ST(H) = \{H_w \mid w \in A^* \wedge |w| \leqslant \log_2 |H|\}$.

Proof. For each $w \in A^*$ and $b \in A$ we either have $H_w = H_{wb}$ or $|H_w| \geqslant 2|H_{wb}|$ because $H_{wb} = H_w \cap H_b$. So, if they are not equal, then their intersection is a subgroup which has index at least 2 in H_w. □

The idea is therefore to use words of length at most $\log_2 |H|$ to represent stabilizers and to perform the calculations for stabilizers on these words. The representation is not unique, but this does not matter for our application.

A main task is to compute a word w of length at most $\log_2 |H|$ such that $H_w = H_u \cap H_v$ (when u and v satisfy $|u|, |v| \leqslant \log_2 |H|$). This can be done efficiently according to the following lemma.

Lemma 3.4. *Every element in the commutative monoid $ST(H)$ of stabilizers can be represented by a word in A^* of length at most* $\log |H|$*, thus with at most* $\mathcal{O}(\log |H| \cdot \log |A|)$ *bits. Using this representation, multiplication (that is, intersection) and computing the H-action (that is, conjugation), can be done in space* $\mathcal{O}(\log |H| \cdot \log |A|)$.

Proof. Let $uv = a_1 \cdots a_m$. We have to compute a word w of length at most $\log_2 |H|$ such that $H_w = H_{a_1 \cdots a_m}$.

We run a loop for $i = 1$ to m. At each step we have computed a word u_{i-1} such that $H_{u_{i-1}} = H_{a_1 \cdots a_{i-1}}$ with the invariant $2^{|u_{i-1}|}|H_{u_{i-1}}| \leqslant |H|$ (initially we let $u_0 = 1$). There are exactly three mutually disjoint cases.

(1) If $H_{u_{i-1}} \subseteq H_{a_i}$ then we let $u_i = u_{i-1}$.
(2) If $H_{a_i} \subsetneq H_{u_{i-1}}$, then we let $u_i = a_i$.
(3) If H_{a_i} and $H_{u_{i-1}}$ are incomparable with respect to containment, then we let $u_i = u_{i-1}a_i$.

Each case keeps the invariant because, by induction, $2^{|u_i|}|H_{u_i}| \leqslant |H|$. □

Remark 3.5. *The reader can easily check that our computation of a word w with $H_w = H_{a_1 \cdots a_m}$ yields a word w of pairwise different letters. So, we could actually put a bound* $|w| \leqslant \min \{\log_2 |H|, |A|\}$ *on its length.*

3.2. ***H*-*N*-monoids.** In the following H denotes a finite group and N denotes a finite H-monoid. Let M be a set (resp. be a monoid) with involution and

$$\mu \colon M \to N$$

be a morphism. We say M (together with μ) is an *H-N-alphabet* (resp. an *H-N-monoid*) if H acts on M such that $\mu(g \cdot x) = g \cdot \mu(x)$. For example, the identity map on N makes N itself to an H-N-monoid.

A morphism between H-N-monoids is an H-compatible morphism $\varphi \colon M' \to M$ such that $\mu\varphi = \mu'$. Thus, if M is an H-N-monoid and M' is a monoid with involution where H acts, then every H-compatible morphism $\varphi \colon M' \to M$ turns M' into an H-N-monoid where μ' is uniquely defined by the equation $\mu\varphi = \mu'$. The use of H-N-monoids is natural in our setting: the H-action is due to a group action on letters, and the finite monoid N is used for the specification of rational constraints. It is clear that the specification of constraints has to be compatible with the group action.

3.3. **Free H-N-monoids and types.** Let B and $\mathcal{Y}$ be two disjoint H-N-alphabets. We call B the alphabet of *constants* and $\mathcal{Y}$ the set of twisted variables. The free monoid with involution $(B\cup\mathcal{Y})^*$ becomes an H-N-monoid where $\mu\colon (B\cup\mathcal{Y})^* \to N$ is induced by $B\cup\mathcal{Y}$.

$$\overline{x_1\cdots x_m} = \overline{x_m}\cdots\overline{x_1},$$
$$g\cdot(x_1\cdots x_m) = g\cdot(x_1)\cdots g\cdot(x_m).$$

By $\theta \subseteq (B\cup\mathcal{Y})^* \times (B\cup\mathcal{Y})^*$ we denote a finite homogeneous relation. Here as usual, a relation is called *homogeneous* if $(x,y)\in\theta$ implies $|x|=|y|$. If $(x,y)\in\theta$ then we also say that (x,y) is a *defining relation* because the algebraic object we are interested in is the quotient monoid

$$(B\cup\mathcal{Y})^*/\left\{x=y \mid (x,y)\in\theta\right\}.$$

We need more structure of this quotient monoid; in particular, $\mu\colon (B\cup\mathcal{Y})^* \to N$ should induce a morphism of H-N-monoids. Actually we wish more, therefore we impose the following technical restrictions on θ; and then we call θ a *type* (and for a variable X we also define the *type of* X denoted $\theta(X)$ below).

(1) $(x,y)\in\theta$ implies $\mu(x)=\mu(y)$, $(\overline{y},\overline{x})\in\theta$, and $(f(x),f(y))\in\theta$ for all $f\in H$, even if these relations are not listed in the specification of θ.
(2) If a (twisted) variable X appears in θ (that is $|xy|_X \geqslant 1$ for some $(x,y)\in\theta$), then we call X *typed*. For a typed variable X we require that there is a unique primitive word[2] $p\in B^*$ such that $(Xp, pX)\in\theta$. We define $\theta(X)=p$, and say that $\theta(X)$ is the *type of* X.
(3) For $(x,y)\in\theta$ we allow exactly three possibilities:
  (i) $(x,y)=(ab,ca)$ with $a,b,c\in B$.
  (ii) $(x,y)=(X\,\theta(X),\,\theta(X)\,X)$ for variables X.
  (iii) $(x,y)=(Xa,aY)$ where $a\in B$ and X,Y are typed variables such that $X\neq Y$.

It is convenient to choose a subset $\mathcal{X}\subseteq\mathcal{Y}$ which is closed under the involution such that every $Y\in\mathcal{Y}$ has the form $Y=f(X)$ for some $X\in\mathcal{X}$ and $f\in H$. In the following, by a *variable* we typically mean $X\in\mathcal{X}$ and thus, every twisted variable $Y\in\mathcal{Y}$ can be written as $f\cdot X$ for some $f\in H$ and $X\in\mathcal{X}$. We assume $X\neq\overline{X}$ for all variables. Having chosen θ and $\mathcal{X}$ we denote by $M(B,\mathcal{X},\theta,\mu)$ the following quotient monoid (and an *H-N-monoid with type* θ):

$$M(B,\mathcal{X},\theta,\mu) = (B\cup\mathcal{Y})^*/\left\{x=y \mid (x,y)\in\theta\right\}.$$

Point (1) from above makes sure that one can extend the involution, the morphism μ and the action of H to the quotient $M(B,\mathcal{X},\theta,\mu)$. The homogeneity condition for θ makes it possible to solve the uniform word problem in $M(B,\mathcal{X},\theta,\mu)$ in nondeterministic quasi-linear space:

Lemma 3.6. *There is an* $\mathsf{NSPACE}(n\log n)$ *algorithm which performs the following task. The input is an alphabet* B*, a homogeneous relation* $\theta\subseteq B^*\times B^*$*, and two words* $u,v\in B^*$ *such that*

$$|uv| + |B| + \sum_{(x,y)\in\theta} |xy| \leqslant n.$$

The output is "yes" if $u=v$ *in the quotient monoid* $B^*/\left\{x=y \mid (x,y)\in\theta\right\}$ *and "no" otherwise.*

[2] Recall that p is primitive if and only if it cannot be written as $p=r^e$ with $e\geqslant 2$.

Proof. If $u = v$ in $B^*/\{x = y \mid (x, y) \in \theta\}$, then nondeterministically we can apply rewriting rules from θ (which preserve length) to u until we see $u = v$ in the free monoid B^*. We get the "no" answer because $\mathsf{NSPACE}(n \log n)$ is closed under complementation by the theorem of Immerman-Szelepcsényi, see for example [42]. □

Using Lemma 3.6 we represent elements in $M(B, \mathcal{X}, \theta, \mu)$ by words over $(B \cup \mathcal{Y})^*$. For $\theta = \emptyset$ we obtain $(B \cup \mathcal{Y})^* = M(B, \mathcal{X}, \emptyset, \mu)$. By $M(B, \theta, \mu)$ we denote the H-N-monoid submonoid with type θ which is generated by B. In particular, $B^* = M(B, \emptyset, \mu)$. If $\theta \cap B^* \times B^* = \emptyset$, then $M(B, \emptyset, \mu) = B^*$ is a free submonoid of $M(B, \mathcal{X}, \theta, \mu)$. If H acts without fixed points on $\mathcal{Y}$, then we identify $\mathcal{Y} = H \times \mathcal{X}$ and the action becomes $g \cdot (f, X) = (gf, X)$. Later we will write typed variables using a special bracket notation $[X, p]$. For complexity issues we will only allow θ which satisfy $|\theta| \in \mathcal{O}(|H| \, \|\mathcal{S}\|^2)$ where $\|\mathcal{S}\|$ is specified in Equation (2) below.

3.4. EDT0L languages and relations.

The acronym EDT0L refers to ***E**xtended, **D**eterministic, **T**able, **0** interaction, and **L**indenmayer.* See the handbook [50] for the many results about **L**-systems. Let A be an alphabet. A subset L in a k-fold direct product $A^* \times \cdots \times A^*$ is called a *EDT0L relation* if there is some (extended) alphabet C with $d_1, \ldots, d_k \in C$ such that $A \subseteq C$ and a rational set $\mathcal{R} \subseteq \mathrm{End}(C^*)$ of endomorphisms over C^* such that

$$L = \{(h(d_1), \ldots, h(d_k)) \mid h \in \mathcal{R}\}.$$

The classical situation refers to $k = 1$. In that case we speak about an EDT0L language; and our definition uses a characterization of EDT0L languages due to [1]. The connection is as follows. Let \$ be a symbol which is not in A and $L \subseteq A^* \times \cdots \times A^*$ be a EDT0L relation, then $\{w_1 \$ w_2 \cdots \$ w_k \mid (w_1, \ldots, w_k) \in L\}$ is an EDT0L language in the usual sense over the alphabet $A \cup \{\$\}$. It should also be noted that the class of EDT0L languages coincides with the class of HDT0L languages ([50, Thm. 2.6]).

We say L is an *effective* EDT0L relation if there is an effective description of an NFA $\mathcal{A}$ with transitions labeled by "deterministic tables" of pairs $(c, u_c) \in C \times C^*$ (encoding the endomorphism which maps c to u_c (and $\overline{c}$ to $\overline{u_c}$))[3] and letters $d_1, \ldots, d_k \in C$ such that $(w_1, \ldots, w_k) \in L$ if and only if there is some $h \in L(\mathcal{A}) \subseteq \mathrm{End}(C^*)$ such that $(w_1, \ldots, w_k) = (h(d_1), \ldots, h(d_k))$.

4. Twisted word equations

4.1. The initial setting.

We begin with a nonempty alphabet of constants A, and a list of $2k$ variables $\mathcal{V}_0$ (as always, both with involution) and a finite group H where H acts on A via a homomorphism $\psi\colon H \to \mathrm{Aut}(A)$. In particular, $|\psi(H)| \leqslant |A|! \leqslant |A|^{|A|}$. As above, H acts on $H \times \mathcal{V}_0$ by $f \cdot (g, X) = (fg, X)$. For $w \in A^*$ and $f \in H$ we also use the notation $f(w) = (f, w)$. Hence, we may represent elements in $(A \cup (H \times \mathcal{V}_0))^*$ by words in $(H \times (A^* \cup \mathcal{V}_0))^*$. We abbreviate $(1, x)$ as x for $x \in A^* \cup \mathcal{V}_0$. By $\mu_0\colon A^* \to N$ we mean a homomorphism which respects the involution and the action of H. Thus A^* is, via μ_0, an H-N-monoid. Assume that μ_0 has been extended to a mapping $\mu_0\colon A^* \cup \mathcal{V}_0 \to N$ such that $\mu_0(\overline{X}) = \overline{\mu_0(X)}$, then μ_0 extends to a morphism $\mu_0\colon (A \cup (H \times \mathcal{V}_0))^* \to N$ of H-N-monoids by $\mu_0(f, X) = f \cdot \mu_0(X)$. Initially we work over free monoids.

Definition 4.1. *A system $\mathcal{S}$ of* twisted word equations with regular constraints *over A and $\mathcal{V}_0$ is given by the following data:*

[3] Without restriction we can assume each transition is labeled by an endomorphism which changes at most one pair of letters $c, \overline{c}$.

- *A list of $2k$ variables such that $\mathcal{V}_0 = \left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}\right\}$.*
- *The set of twisted variables becomes $\mathcal{Y} = H \times \mathcal{V}_0$.*
- *A set of pairs $\{(U_i, V_i) \mid 1 \leqslant i \leqslant s\}$ where $U_i, V_i \in (A \cup \mathcal{Y})^*$.*
- *A morphism $\mu_0 \colon (A \cup \mathcal{Y})^* \to N$ of H-monoids, with N finite.*

A solution *of $\mathcal{S}$ is a morphism of sets with involution $\sigma \colon \mathcal{V}_0 \to A^*$ which is (uniquely) extended to an A-morphism of H-N-monoids $\sigma \colon (A \cup \mathcal{Y})^* \to A^*$ such that*

- *$\sigma(U_i) = \sigma(V_i)$ for all pairs (U_i, V_i).*
- *$\mu_0\sigma(X) = \mu_0(X)$ for all variables. Hence, $\mu_0\sigma = \mu_0$.*

As usual, a pair (U_i, V_i) representing a twisted equation is simply written as $U_i = V_i$.

Example 4.2. *Let $A = \{a, \overline{a}, b, \overline{b}\}, \mathcal{V}_0 = \{X, \overline{X}, Y, \overline{Y}, Z, \overline{Z}\}$, $f, g \in H$ defined by $f(a) = b, g(a) = \overline{a}, g(b) = b$, $U_1 = (f, X)a(g, \overline{Y})$, $V_1 = Z$, $U_2 = (f, Y)b$, $V_2 = \overline{a}b(g, X)$, $U_3 = Xa$, $V_3 = b(f, X)$ and (for simplicity) $\mu_0(x) = 1$ for all $x \in A \cup \mathcal{V}_0$. A pictorial representation of the example is shown in Figure 2. The reader is invited to verify that one possible solution is $\sigma(X) = bab, \sigma(Y) = \overline{b}aa\overline{b}, \sigma(Z) = abaabaab$, and a second solution is $\sigma(X) = b, \sigma(Y) = \overline{b}a, \sigma(Z) = aaab$.*

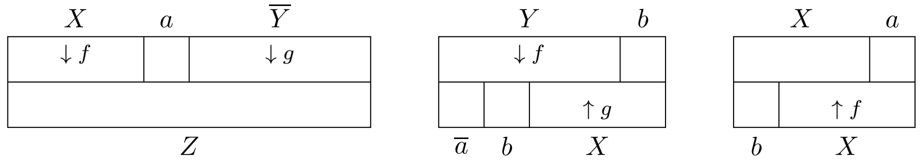

FIGURE 2. Pictorial representation of Example 4.2.

If σ is a solution of $\mathcal{S}$ we also say that σ *solves* $\mathcal{S}$. For $\left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}\right\}$ the *full solution set* $\mathrm{Sol}(\mathcal{S})$ of $\mathcal{S}$ is defined as

$$\mathrm{Sol}(\mathcal{S}) = \{(\sigma(X_1), \ldots, \sigma(X_k)) \in A^* \times \cdots \times A^* \mid \sigma \text{ solves } \mathcal{S}\}.$$

4.2. **The main result on twisted word equations.** Our main result shows that $\mathrm{Sol}(\mathcal{S})$ is an EDT0L language, for which we can compute an effective description in polynomial space. In order to measure complexities accurately, we need a precise notion of input size. Let $\mathcal{S}$ be a system of twisted word equations with regular constraints over A and $\mathcal{V}_0$ according to Definition 4.1.

We define the size of $\mathcal{S}$ using two parameters $\|\mathcal{S}\|$ and $m(\mathcal{S})$. Thus, the size is the pair $(\|\mathcal{S}\|, m(\mathcal{S}))$. The first parameter $\|\mathcal{S}\|$ ignores the size of the finite monoid N. It is the main parameter as we don't want that the complexity due to constraints dominates the overall complexity. The second parameter measures separately the number of additional bits for handling the constraints. We begin by defining $\|\mathcal{S}\|$. Let

$$\|\mathcal{S}\| = |H| + |A| + k + s + \sum_{1 \leqslant i \leqslant s} |U_iV_i|\,. \tag{2}$$

Recall that $2k$ is the number of variables, s is the number of equations $U_i = V_i$, and H denotes a finite group acting on A and hence on A^*, too. We are interested in a situation only where $A \neq \emptyset$ and $\|\mathcal{S}\| > 4$.

The finite monoid N is also part of the input. We measure the size relative to $\|\mathcal{S}\|$. Let N be any finite H-monoid and $m_{\mathcal{S}}(N) \in \mathbb{N}$ be a number such that elements of N can be encoded by a number of bits which is at most

$$m_{\mathcal{S}}(N)(2 + \log|A|) \cdot \log\|\mathcal{S}\|\,. \tag{3}$$

Moreover, using this specification, monoid computations, like computing the involution, the multiplication of two elements, and the action by H, can be performed on a Turing machine in space $m_{\mathcal{S}}(N)\,\|\mathcal{S}\|\log\|\mathcal{S}\|$. We let

$$m(\mathcal{S}) = m_{\mathcal{S}}(N) \text{ where } N \text{ is the monoid which appears in } \mathcal{S} \tag{4}$$

There are examples where the monoid N (which appears in $\mathcal{S}$) is polynomially bounded in $\|\mathcal{S}\|$ and still $m(\mathcal{S}) \in \mathcal{O}(1)$. However, if $m(\mathcal{S})$ becomes a polynomial in $\|\mathcal{S}\|$, then we need to consider $m(\mathcal{S})$ separately for a finer analysis below PSPACE.

An H-monoid N is called *small with respect to* $\mathcal{S}$ if $m_{\mathcal{S}}(N) \in \mathcal{O}(1)$. The finite monoid recognizing reduced words (those without factors $a\overline{a}$ for $a \in A$) is small with respect to $\mathcal{S}$, see Example 2.1. Being small is no restriction for the computability of the representation of the full solution set as an EDT0L relation – we can always add trivial equations until N becomes small with respect to $\mathcal{S}$. Another example of a small monoid is the finite monoid $\mathrm{ST}(H)$ of stabilizers. Its size depends on A and H, but still it is small due to Lemma 3.4.

Now, during the process we might wish to use direct products of (small) monoids. For that the parameter $m_{\mathcal{S}}$ behaves nicely:

$$m_{\mathcal{S}}(N_1 \times N_2) \leqslant m_{\mathcal{S}}(N_1) + m_{\mathcal{S}}(N_2).$$

Indeed, given $(n_1, n_2) \in N_1 \times N_2$ we can use the first $m_{\mathcal{S}}(N_1)$ bits to encode n_1 and the last $m_{\mathcal{S}}(N_2)$ bits to encode n_2. The operations on $N_1 \times N_2$ can be done component wise. In particular, a direct product of small monoids remains small.

We are ready to state our main result which gives PSPACE as an upper bound for the complexity and a quasi-quadratic space bound if N is small.

Theorem 4.3. *There is an* $\mathsf{NSPACE}(|H| \cdot \|\mathcal{S}\|^2 \cdot m(\mathcal{S}) \cdot \log|A| \cdot \log\|\mathcal{S}\|)$ *algorithm which performs the following task. It takes as input a system of twisted word equations* $\mathcal{S}$ *with regular constraints. The system use a set of constants* A*, a set of variables* $\mathcal{V}_0 = \left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}\right\}$*, and the regular constraint is given by a morphism* $\mu_0 \colon A \cup \mathcal{V}_0 \to N$*. The output is:*

- *an extended alphabet* C *of size* $\mathcal{O}(|H|^2\,\|\mathcal{S}\|^2)$*;*
- *distinguished letters* $d_i \in C$ *for each variable* X_i*;*
- *a trim NFA* $\mathcal{A}_{\mathcal{S}}$ *accepting a rational set of* A*-morphisms* $L(\mathcal{A}_{\mathcal{S}}) \subseteq \mathrm{End}(C^*)$ *such that*

$$\mathrm{Sol}(\mathcal{S}) = \{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{\mathcal{S}})\}\,. \tag{5}$$

The algorithm stores intermediate equations with a length bound in $\mathcal{O}(|H|\,\|\mathcal{S}\|^2)$*. Moreover,* $\mathrm{Sol}(\mathcal{S}) = \emptyset$ *if and only if* $\mathcal{A}_{\mathcal{S}} = \emptyset$*; and* $|\mathrm{Sol}(\mathcal{S})| < \infty$ *if and only if* $\mathcal{A}_{\mathcal{S}}$ *doesn't contain any directed cycle.*

Let us comment on the rather complicated space bound

$$|H| \cdot \|\mathcal{S}\|^2 \cdot m(\mathcal{S}) \cdot \log|A| \cdot \log\|\mathcal{S}\|$$

which appears in the statement of the theorem. First, since $|H| \leqslant \|\mathcal{S}\|$ and $|C| \in \mathcal{O}(|H|^2\,\|\mathcal{S}\|^2)$, we can encode all letters by $\mathcal{O}(\log\|\mathcal{S}\|)$ bits. Second, the μ-value for the constraints changes dynamically: it is a priori not fixed for the extended alphabet C. So, it is enough to store the μ-value for each symbol which appears in intermediate equations. The length bound on intermediate equations is in $\mathcal{O}(|H|\,\|\mathcal{S}\|^2)$. Each μ-value is an element in N, which requires, by definition, $m(\mathcal{S}) \cdot \log|A| \cdot \log\|\mathcal{S}\|$

bits for the encoding. Together, we need $\mathcal{O}(|H| \cdot \|\mathcal{S}\|^2 \cdot m(\mathcal{S}) \cdot \log |A| \cdot \log \|\mathcal{S}\|)$ bits to encode intermediate equations,

Corollary 4.4. *Let $\mathcal{S}$ be a system of twisted word equations with regular constraints in variables $\left\{X_1, \overline{X_1}, \dots, X_k, \overline{X_k}\right\}$. Then $\mathrm{Sol}(\mathcal{S}) \subseteq A^* \times \cdots \times A^*$ is an effective EDT0L relation.*

Proof. This is a formal consequence of Theorem 4.3. □

Sections 5 through 12 are devoted to the proof of Theorem 4.3. The theorem implies that we can decide in PSPACE (hence, in deterministic singly exponential time) whether $\mathcal{S}$ is solvable and whether or not there are only finitely many solutions. The decision problem of whether a word equation with regular constraints has a solution is known to be PSPACE-hard by [31] because the intersection problem of regular languages is a special case. In our setting, if the finite monoid N is small, then the best known lower bound to date is NP-hardness: it is the lower bound for deciding whether or not a linear Diophantine system over $\mathbb{N}$ has a solution [24].

5. Preparation

We begin the proof of Theorem 4.3 with some technical preparations. Sections 5.1–5.3 concern some reductions, and could easily be skipped in a first reading of the paper. These sections yield a reduction to the situation as stated in beginning of Section 5.4. We invite the reader to jump directly to Section 5.4 and to read the parts in between only when necessary.

5.1. **Reducing to faithful actions.** Recall that the action of H on A is given by a homomorphism $\psi\colon H \to \mathrm{Aut}(A)$. We don't require that ψ is injective because in some natural examples this is not the case, see Section 15. On the other hand it is enough to prove Theorem 4.3 in the case where H is actually a subgroup of $\mathrm{Aut}(A)$. Let us show how the reduction works. The principal idea is to replace H by H/K where $K = \ker(\psi)$ is the kernel of ψ.

If M is any H-monoid, then the action of H induces an action of H/K on M only if for all $f \in K$ and all $m \in M$ we have $f(m) = m$. In this case M becomes an H/K monoid: the action $g \cdot K(m) = g(m)$ is well-defined for all $g \cdot K \in H/K$. By definition of K, the free monoid A^* is therefore an H/K-monoid. Inspecting the statement in Theorem 4.3, there are two problems: the induced action of K on the finite monoid is not trivial, in general. Moreover, the group acts H freely on the set of variables $H \times \mathcal{X}_0$, so there is no induced action of H/K on this set unless K is trivial. We address and solve both problems.

Let us begin with the H-monoid N, then it has a largest H-invariant submonoid N' where K acts trivially. It is the submonoid of K-invariant elements:

$$N' = \{m \in N \mid \forall f \in K : \, f(m) = m\}\,.$$

The image $\mu_0(A^*)$ is a submonoid of N', since for all $f \in K$ and all $w \in A^*$ we have $f(\mu_0(w)) = \mu_0(f(w)) = \mu_0(w)$. However, the statement in Theorem 4.3 doesn't require that $\mu_0(X)$ takes values in N'. Let us show that $\mathcal{S}$ is not solvable if there is some variable X such that $\mu_0(X) \notin N'$. Indeed, assume the contrary that there is a solution $\sigma\colon \mathcal{X}_0 \to A^*$ such that $\mu_0(X) \notin N'$. Then there is some $f \in K$ such that

$$\mu_0(X) \neq f(\mu_0(X)) = f(\mu_0\sigma(X)) = \mu_0(f(\sigma(X))) = \mu_0\sigma(X) = \mu_0(X),$$

which is a contradiction. We have $f(\sigma(X)) = \sigma(X)$ because K acts trivially on A^*.

Thus, as a first procedure in the proof of Theorem 4.3 we check that $\mu_0(X) \in N'$ for all $X \in \mathcal{X}_0$ (and therefore $\mu_0(f, X) \in N'$ for all $(f, X) \in H \times \mathcal{X}_0$). The test runs over all $f \in H$ and for each f checks the following implication:

$$(\forall a \in A : f(a) = a) \implies f(\mu_0(X)) = \mu_0(X). \tag{6}$$

If the check is positive then $\mu_0(X) \in N'$ for all $X \in \mathcal{X}_0$. If the check fails then we output that $\mathcal{S}$ is not solvable by defining $\mathcal{A}_{\mathcal{S}} = \emptyset$. We can perform the test within our given space bound by definition of $m(\mathcal{S})$.

After the test, we may assume that μ_0 maps $A \cup (H \times \mathcal{X}_0)$ to N', and we replace N by N'. We can use the same bit encoding for elements in N' as we did for N, but if we have to guess an element in N', we perform the test (6). Thus, the parameter $m(\Phi)$ is still valid.

In the second step we replace each variable $(f, X) \in H \times \mathcal{X}_0$ by a fresh variable $(f \cdot K, X) \in (H/K) \times \mathcal{X}_0$. Again, this doesn't change $\mathrm{Sol}(\mathcal{S})$ since for every H-compatible morphism $\sigma\colon H \times \mathcal{X}_0 \to A^*$ and all $f \in K$ we have

$$\sigma(f, X) = f(\sigma(1, X)) = f(\sigma(X)) = \sigma(X) \text{ and } \mu_0(f, X) = f(\mu_0(X)) = \mu_0(X).$$

We are done. We have shown the following statement.

Lemma 5.1. *It is enough to prove Theorem 4.3 under the additional assumption that ψ is injective. This means we can assume that H is a subgroup of* $\mathrm{Aut}(A)$.

5.2. **Making the finite monoid N larger.** The aim in this section is to replace N by a larger monoid, which additionally encodes information about stabilizers for all $x \in A \cup \mathcal{X}_0$. Up to a constant factor we don't change $m(\mathcal{S})$.

Let $\mathrm{ST}(H)$ be the monoid of stabilizers, see Section 3.1. We define a morphism $\mu_1\colon A \cup \mathcal{X}_0 \to N \times \mathrm{ST}(H)$ which maps a letter $a \in A$ to $(\mu_0(a), H_a)$ and each variable X to some $(\mu_0(X), H_u)$ where $u \in A^*$ is any word of length at most $\log_2 |H|$ by guessing u. The H-action on $N \times \mathrm{ST}(H)$ is inherited from the action on N and the action on $\mathrm{ST}(H)$ by conjugation. Moreover, guessing is equivalent to taking the union over finitely many cases, see (7). The union will give the same solutions we had before, and it will not introduce any new solutions.

The projection to the first component turns $N \times \mathrm{ST}(H)$ into an H-N-monoid. Using μ_1 we achieve the following:

- for all $x \in (A \cup \mathcal{X}_0)^*$, $\mu_1(x) \in N \times \mathrm{ST}(H)$ is a pair where the second component is H_x which is represented by a word $u \in A^*$ of length at most $\log |H|$ such that $H_x = H_u$.

The switch to μ_1 has a price. By defining $\mu_1(X)$ we restrict the set of possible solutions. The value $\mu_1(X) = (\mu_0(X), H_u)$ fixes the stabilizer $H_{\sigma(X)}$ for a solution σ to be the subgroup H_u. The number of choices (= nondeterministic guesses) to extend μ_0 to μ_1 is bounded by

$$(|\mathrm{ST}(H)|)^k \leqslant |A|^{|\mathcal{X}_0| \log |H|}. \tag{7}$$

These choices result in a splitting the original system into that many subsystems. Splitting is fortunately no problem since EDT0L languages are closed under finite union by taking the unions of the corresponding NFAs.

At the end of this we rename $N \times \mathrm{ST}(H)$ as N and μ_1 as $\mu_0\colon A \cup \mathcal{X}_0 \to N$.

5.3. **Introducing a zero to N and a marker symbol to A.** In the following it is convenient to have a special symbol #, but we want to make sure no variable uses it, so we add 0 to our constraint monoid. We next embed our current N into $N \cup \{0\}$ where 0 is a fresh symbol not included in N and 0 acts as a zero in $N \cup \{0\}$. We turn it into an H-monoid by defining $f(0) = 0$ for all $f \in H$.

The monoid N is an H-submonoid of $N \cup \{0\}$ and, by a slight abuse of language, we denote by μ_0 the induced mapping to the larger monoid $N \cup \{0\}$ as well:xt

$$\mu_0\colon A \cup (H \times \mathcal{X}_0) \xrightarrow{\mu_0} N \hookrightarrow N \cup \{0\}\,.$$

Without restriction (by adjusting constants if necessary) we may assume that $N \cup \{0\}$ doesn't change the parameter $m(\mathcal{S})$. Using $N \cup \{0\}$ instead of N doesn't change $\mathrm{Sol}(\mathcal{S})$ because $\mu_0(A \cup (H \times \mathcal{X}_0)) \subseteq N$. Phrased differently, without restriction N has a zero 0; and $\mu_0(A \cup (H \times \mathcal{X}_0)) \subseteq N \setminus \{0\}$.

At this point we to add the special symbol # to A. We let $\mu_0(\#) = 0$, $\overline{\#} = \#$ and $f(\#) = \#$ for all $f \in H$. So, from now on we assume that $\# \in A$. Since we did not change $\mu_0(X)$ for any variable X we are sure that for every solution σ to $\mathcal{S}$ and every variable X we have $|\sigma(X)|_{\#} = 0$: the marker cannot appear in any solution.

5.4. **Triangular systems.** Due to the preceding subsections we henceforth make the following assumptions:

- H is a subgroup of $\mathrm{Aut}(A)$.
- There is some $\# \in A$ with $\overline{\#} = \#$ and $f(\#) = \#$ for all $f \in H$.
- The H-monoid N contains a zero 0 and for all $x \in A \cup \mathcal{X}_0$ we have
$$\mu_0(x) = 0 \iff x = \#.$$
- $\mu_0(x)$ is a pair where the second component is the stabilizer H_x which is represented by a word $u \in A^*$ of length at most $\log |H|$ such that $H_x = H_u$ for all $x \in (A \cup \mathcal{X}_0)^*$. Since $H_x = H_u$ and u is in A^* we have for all $f \in H$:

$$f \in H_x \iff f(u) = u. \tag{8}$$

Definition 5.2. *A twisted word equation $U = V$ is called* triangular *if U contains at most two variables and V at most one variable.*

Following well-known methods (see for example [8]) we enlarge the set of variables $\mathcal{X}_0$ to a larger set $\mathcal{X} \supset \mathcal{X}_0$ (using at most $2\,\|\mathcal{S}\|$ more variables) such that every equation becomes triangular in a more specific form: every equation has the form either $Z = 1$ or $U = Z$ where $|U| = 2$ and in both cases Z is a variable.

It therefore is enough to show Theorem 4.3 in the case where each each equation $U_i = V_i$ equals $(f, x)(g, y) = (h, Z)$ where $x, y \in A \cup \mathcal{X}$ and $Z \in \mathcal{X}$. Moreover, since $(f, x)(g, y) = (h, Z)$ is equivalent to $(h^{-1}f, x)(h^{-1}g, y) = Z$ we can restrict ourselves to the case that each $U_i = V_i$ is of the form $(f, x)(g, y) = Z$. Due to additional variables, we work over a set of variables

$$\mathcal{X} = \left\{X_1, \overline{X_1}, \dots, X_k, \overline{X_k}, X_{k+1}, \overline{X_{k+1}}, \dots, X_{k'}, \overline{X_{k'}}\right\}$$

where $k \leqslant k' \leqslant 2\,\|\mathcal{S}\|$ and the first $2k$ variables belong to the original system.

Hence, the starting point is a system of equations $(f, x)(g, y) = Z$. The number of these triangular equations is at most $2\,\|\mathcal{S}\|$, so we can ignore this blow-up. During the process we need a more general form, nontrivial triangular equations appear as $u(f, x)w(g, y)v = u'Zv'$ where u, w, v, u', v' are words over constants. Whenever such an equation with $|u| = |u'| = |v| = |v'|$ appears, then necessarily $u = u'$ and $v = v'$; otherwise the equation is "unsolvable". That is, in a nondeterministic implementation of our process, this branch never leads to an accepting state. In an implementation of the algorithm we would reject the branch immediately.

Finally, it is somewhat convenient to assume $|A| + |\mathcal{X}| \leqslant |UV|$. We may achieve this for example by adding some dummy equations, and then $\|\mathcal{S}\| \in \mathcal{O}(|H| + |UV|)$.

5.5. **Fixing more notation.** During the process we enlarge the sets of constants and variables. We begin with two disjoint infinite alphabets with involution C and Ω and $\Sigma = C \cup \Omega$. All constants are drawn from C and all variables are drawn from Ω. We never write down all elements from C or Ω, just certain subsets which are needed in a specific situation. Later we will choose Σ such that $|\Sigma| \in \mathcal{O}(|H|^2\,\|\mathcal{S}\|^2)$, but initially for our infinite automata $\mathcal{T}$ and $\mathcal{F}$ we do not impose any size restrictions.

Throughout we use following conventions and notation.

- There are $2k$ *distinguished letters* $\left\{d_1, \overline{d_1}, \ldots, d_k, \overline{d_k}\right\}$ which appear in Theorem 4.3.
- $\# \in A \subseteq B = \overline{B} = \{f(b) \mid b \in B\} \subseteq C$.
- $B \cap \left\{d_1, \overline{d_1}, \ldots, d_k, \overline{d_k}\right\} = \emptyset$ unless we are at a *final state* (to be defined below, see Section 7.1.2).
- $\mathcal{Y} = \overline{\mathcal{Y}} = \{f(X) \mid X \in \mathcal{X}, f \in H\} \subseteq \Omega$, and $X \neq \overline{X}$ for all $X \in \Omega$. If H acts freely on $\mathcal{Y}$, then we write $\mathcal{Y} = H \times \mathcal{X}$, too. We view $\mathcal{X} \subseteq \mathcal{Y}$.
- The action of H and the involution on Σ extend those on $A \cup \mathcal{Y}$.
- $\mu\colon (B \cup \mathcal{Y})^* \to N$ satisfies $\mu(a) = \mu_0(a)$ for $a \in A$.
- $a, b, c, [p], [r, s, \lambda], \ldots$ refer to letters in C.
- $u, v, w, \ldots$ refer to words in C^*.
- $X, Y, Z, [X, p], \ldots$ refer to variables in Ω.
- $x, y, z, \ldots$ refer to words in Σ^*.

These conventions hold everywhere unless explicitly stated otherwise. They also apply to primed symbols such as B', $\mathcal{X}'$ etc. Throughout we also use the following.

Remark 5.3. *If we know $\mu(x) \in N$ for any $x \in (B \cup \mathcal{Y})^*$, then we also know, by the representation of the second component in $\mu(x)$, a word $u \in A^*$ of length at most $\log|H|$ such that $H_x = H_u$. This enables for all $f \in H$ an efficient test to check whether $f(x) = x$, see (8) above. Moreover, if we have $z = xy$ with $x, y, z \in (B \cup \mathcal{Y})^*$ and we have to calculate $\mu(z)$ as the product $\mu(x)\mu(y)$, then we need to find a word $w \in A^*$ of length at most $\log|H|$ such that $H_z = H_x \cup H_y = H_w$. We may assume that H_x and H_y are already given as $H_x = H_u$ and $H_y = H_v$ where $uv \in A^*$ of length at most $2\log|H|$. In order to compute w we run the algorithm from the proof of Lemma 3.4.*

5.6. **The initial word equation W_{init}.** For technical reasons we encode the initial (triangular) system $\{(U_i, V_i) \mid 1 \leqslant i \leqslant s\}$ of twisted equations in variables $\mathcal{X}$ where $\{X_i \mid 1 \leqslant i \leqslant |\mathcal{X}|/2\} \subseteq \mathcal{X} = \overline{\mathcal{X}}$ as a single word. Let $U = U_1 \# U_2 \cdots \# U_s$ and $V = V_1 \# V_2 \cdots \# V_s$.

The *initial equation* $W_{\text{init}} \in (A \cup (H \times \mathcal{X}))^*$ is defined as:

$$W_{\text{init}} = \# X_1 \cdots \# X_{|\mathcal{X}|/2} \# U \# \# \overline{V} \# \overline{X_{|\mathcal{X}|/2}} \# \cdots \overline{X_1} \#. \tag{9}$$

In particular, each $X \in \mathcal{X}$ appears in W_{init}. Here:

$$\begin{aligned} \mathcal{X}_0 &= \left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}\right\} \text{ and} \\ \mathcal{X} &= \left\{X_1, \overline{X_1}, \ldots, X_k, \overline{X_k}, X_{k+1}, \overline{X_{k+1}}, \ldots, X_{|\mathcal{X}|/2}, \overline{X_{|\mathcal{X}|/2}}\right\} \end{aligned}$$

Note that $\sigma(W) = \sigma(\overline{W})$ if and only if $\sigma(U_i) = \sigma(V_i)$ for all i.

5.7. **Fixing the parameters n, ε, and δ.** Having defined W_{init} we fix the following parameters $n, \varepsilon, \delta \in \mathbb{N}$ by

$$n = |W_{\text{init}}|, \quad \varepsilon = 30n, \quad \text{and } \delta = |H|\,\varepsilon = 30|H|n. \tag{10}$$

By our assumptions this implies $n > |A| + |\mathcal{X}|$. We have $\|\mathcal{S}\| \in |H| + \Theta(n)$. Moreover, since $n \in \mathcal{O}(\|\mathcal{S}\|)$, we have $\delta \in \mathcal{O}(|H|\,\|\mathcal{S}\|)$ and $\varepsilon \in \mathcal{O}(\|\mathcal{S}\|)$. As a consequence:

$$\mathcal{O}(\delta n) \subseteq \mathcal{O}(|H|\,\|\mathcal{S}\|^2). \tag{11}$$

Note that the right-hand side in (11) coincides with the space bound we allow to store intermediate equations according to Theorem 4.3.

5.8. **Extended equations and their solutions.** The NFA over the monoid $\mathrm{End}(C^*)$ we will construct uses extended equations as states. The overall strategy is to remove variables from the equation until no variables remain. During the process we will enter a phase called δ-periodic compression repeatedly (which is the analogue of "block compression" for solving word equations in free groups). During each call of δ-periodic compression, each variable may create temporarily two new variables which will vanish before the end of that call. So at most $3\,|H|\,n$ variables are needed, including these additional (temporary) variables.

Definition 5.4. *An* extended equation *is a tuple $E = (W, B, \mathcal{X}, \theta, \mu)$, where $A \subseteq B$ and $M(B, \mathcal{X}, \theta, \mu)$ is an H-N-monoid with type θ. Moreover, we require:*

(1) $W \in M(B, \mathcal{X}, \theta, \mu)$ which can be written as a word in the form:

$$W = \#x_1\# \cdots \#x_{|\mathcal{X}|/2}\#u_1 \cdots \#u_s\#\,\#\overline{v_s}\# \cdots \overline{v_1}\#\overline{x_{|\mathcal{X}|/2}}\# \cdots \overline{x_1}\#$$

with x_i, u_j, $v_k \in (B \cup \mathcal{Y})^*$ and $\mu(x_i) \neq 0, \mu(u_j) \neq 0, \mu(v_k) \neq 0$.

(2) Given W as above we call $u_i = v_i$ a local equation*.*

(3) $|W|_\# = |W_{\mathrm{init}}|_\#$.

(4) For every $X \in \mathcal{X}$ there exists some $f \in H$ such that $f(X)$ appears in W.

(5) We say E is a standard state *if first, $\theta = \emptyset$, and second, all local equations are triangular.*

(6) If $E = (W, B, \mathcal{X}, \emptyset, \mu)$ is a standard state, then $\mathcal{X} \subseteq \mathcal{X}$. Moreover,

$$\sum_{Y \in \mathcal{Y}} |W|_Y \leqslant \sum_{Y \in \mathcal{X}} |W_{\mathrm{init}}|_Y.$$

(7) If $E = (W, B, \mathcal{X}, \theta, \mu)$ is any state, then $\sum_{X \in \mathcal{X}} |W|_X \leqslant 3n$. (Thus, we can bound Ω by $|\Omega| \leqslant 3|H|n$ right away.)

(8) If variables X, Y are typed with $X \neq Y$ and $(Xa, aY) \in \theta$, then we have $\theta(X)a = a\theta(Y)$ in the submonoid $M(B, \theta, \mu)$ generated by B.

Recall (Section 3.3) that if a variable X is typed, then there is a primitive word $p = \theta(X)$ such that $(Xp, pX) \in \theta$.

Definition 5.5. *Let $E = (W, B, \mathcal{X}, \theta, \mu)$ be an extended equation.*

- *A* solution *is a B-morphism $\sigma \colon M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ such that:*
  - *$\sigma(W) = \sigma(\overline{W})$.*
  - *$\sigma(X) \in p^*$, whenever X is typed and $p = \theta(X)$.*
- *An* entire solution *is a pair (α, σ) where $\alpha \colon M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ is an A-morphism and σ is a solution.*

6. Twisted conjugacy and δ-periodic words

A key step in proving Theorem 4.3 is to solve a particular kind of a twisted equation: conjugacy. Let $x, y, z \in A^*$. An easy exercise in combinatorics on words [23] yields:

$$zy = xz \iff \exists r, s \in A^* \, \exists e \in \mathbb{N} : x = rs \wedge y = sr \wedge z = (rs)^e r. \tag{12}$$

This fact is crucial in Makanin's classical approach [36] to solve (untwisted) word equations. Here, we need a variant of (12) in the twisted environment. We say that words $x, y \in A^*$ are *twisted conjugate* if there are $f, g, h \in H$ and $z \in A^*$ such that $zg(y) = h(x)f(z)$. We also say that $|x| = |y|$ is the *offset* of the conjugacy. A *twisted conjugacy equation* is a (non-triangular) twisted equation of the form

$$Z\,(g, Y) = (h, X)\,(f, Z). \tag{13}$$

Proposition 6.1. *Let σ be a solution of the twisted equation (13) such that the offset $|\sigma(X)|$ satisfies $1 \leqslant |\sigma(X)| < |\sigma(Z)|$. Then there are words $r \in A^+$, $s \in A^*$ and $e, j \in \mathbb{N}$ with $0 \leqslant j < |H|$ such that $|rs| = \sigma(X)$ and*

$$\sigma(Z) = ((rs)f(rs) \cdots f^{|H|-1}(rs))^e \, f^0(rs) \cdots f^{j-1}(rs)f^j(r). \tag{14}$$

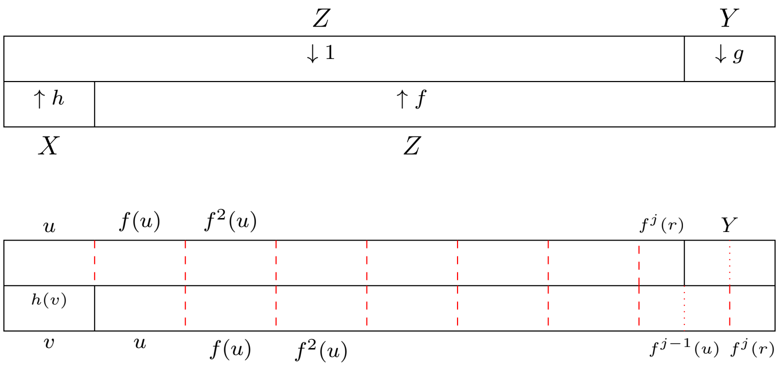

FIGURE 3. Twisted conjugacy

Proof. Let $v = \sigma(X)$ and $u = h(v)$. Since $1 \leqslant |\sigma(X)| < |\sigma(Z)|$ the word u is a proper nonempty prefix of $\sigma(Z)$. If $2\,|u| \leqslant |\sigma(Z)|$, then $uf(u)$ is a prefix of $\sigma(Z)$, and so on. Thus, $\sigma(Z)$ is a prefix of a word $uf(u)f^2(u) \cdots f^k(u)$ for some $k \in \mathbb{N}$. Next, observe that $f^{|H|}(u) = f^0(u) = u$ for every word $u \in A^*$. Thus,

$$\sigma(Z) = [uf(u)f^2(u) \cdots f^{|H|-1}(u)]^e uf(u) \cdots f^{j-1}(u)f^j(r)$$

where $0 \leqslant j < |H|$, $u = rs$ and the $|r|$ suffix of Z is where the pattern runs out, as illustrated in Figure 3. We then have $\sigma(Y) = g^{-1}f^j(sf(r))$. Hence, the nonempty word u and the length $|\sigma(Z)|$ define a unique factorization $u = rs$, integers $0 \leqslant e$ and $0 \leqslant j < |H|$ such that $\sigma(Z)$ has the desired form above. □

A word p is called *primitive* if it cannot be written as $p = r^e$ with $e \geqslant 2$. In particular, the empty word 1 is not primitive. It is well known (and easy to see) that a nonempty word p is if and only if p^2 cannot be written as $p^2 = xpy$ with $x \neq 1$ and $y \neq 1$.

Let $w, p \in A^+$ be nonempty words. We say that *w has period $|p|$* if w is a prefix of $p^{|w|}$. In other words, if $w = a_1 \cdots a_n$ with $a \in A$, then $a_i = a_{i+|p|}$ for all $1 \leqslant i \leqslant n - |p|$. A word may have several periods, for example $w = aabaabaa$ has periods $3, 6, 7, 8$. If $|p|$ is the least period of w, then $|p| \leqslant |w|$ and we can choose p to be primitive such that $p \leqslant w$. For example, $aab \leqslant aabaabaa$ is a primitive prefix and $|aab| = 3$.

Corollary 6.2. *Let $\varepsilon \in \mathbb{N}$, $f, g, h \in H$, and $x, y, z \in A^*$ be words with $1 \leqslant |x| \leqslant \varepsilon$ and $|z| \geqslant |H|\,\varepsilon$. If we have $zg(y) = h(x)f(z)$, then z has a period of at most $|H|\,\varepsilon$.*

Moreover, let $z = \alpha w \beta$ be any factorization with $|w| = |x|$. Then every letter b occurring in z satisfies $b = f(a)$ for some $f \in H$ and some letter a occurring in w.

Proof. By Proposition 6.1 we have

$$z = ((rs)f(rs) \cdots f^{|H|-1}(rs))^e \, f^0(rs) \cdots f^{j-1}(rs)f^j(r)$$

where $|f^i(rs)| = |x| \leqslant \varepsilon$ for all $i \geqslant 0$. Hence, z has a period

$$|(rs)f(rs) \cdots f^{|H|-1}(rs)| \leqslant |H|\,\varepsilon.$$

For the second claim, if $z = \alpha w \beta$ with $|w| = |x| = |rs|$ then w is a factor of $f^i(rs)f^{i+1}(rs)$ for some $i \geqslant 0$. If we write $rs = a_1 \dots a_{|x|}$, then any letter b in z satisfies $b = f^j(a_\ell)$. Let $\iota \in \{i, i+1\}$ so that $f^\iota(a_\ell)$ is a letter in w, then $b = f^j(a_\ell) = f^j(f^{-\iota}(f^\iota(a_\ell))) = f^{j-\iota}(a)$ for some $j \geqslant 0$. □

Definition 6.3. *We say that a word w is δ-*periodic *if it has some period of length at most δ. A δ-periodic word w is called* long δ-periodic *if $|w| \geqslant 3\delta$, and* very long δ-periodic *if $|w| \geqslant 10\delta$.*

For example, $aabaaabaaab$ is 4-periodic but not long 4-periodic. An important property of δ-periodic words is the following.

Lemma 6.4. *Let w be a δ-periodic word and $w = p^e r = q^f s$ such that p, q are primitive $|p| \leqslant |q| \leqslant \delta$, $1 \neq r \leqslant p$, $1 \neq s \leqslant q$, and $|w| \geqslant 2\delta$. Then $p = q$, $e = f \geqslant 1$, and $r = s$.*

Proof. The assertion is clear for $|p| = |q|$. Hence we may assume that p is a proper prefix of q. Since $q \leqslant w$ we conclude $q \leqslant p^\delta$. Since $|w| \geqslant 2\delta$, and $|p| \leqslant |q| \leqslant \delta$ we see $pq \leqslant w \leqslant q^{|w|}$. Thus q occurs as a factor inside qq: we have $pqs = qq$ for some s. Since $1 \leqslant |p| < |q|$, this contradicts the primitivity of q. □

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 see a δ-periodic word.

7. The ambient infinite automaton $\mathcal{T}$

The states of the NFA $\mathcal{A}_\mathcal{S}$ (we are aiming for in Theorem 4.3) are extended equations and transitions are certain labeled arcs between states which modify the extended equations. Before we construct $\mathcal{A}_\mathcal{S}$ let us define an infinite automaton $\mathcal{T}$. It will contain $\mathcal{A}_\mathcal{S}$ as a finite subautomaton. We show that $\mathcal{T}$ is *sound*: this means in the notation of Theorem 4.3

$$\{(h(d_1), \dots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{T})\} \subseteq \mathrm{Sol}(\mathcal{S}). \tag{15}$$

This implies that all subautomata of $\mathcal{T}$ are sound, too. The set of states in $\mathcal{T}$ is the set of extended equations according to Definition 5.4, see Section 7.1. There are two kinds of transitions: a *substitution transition* transforms the variables; a *compression transition* affects the constants, but not the variables, see Section 7.2.

If $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$ is a transition, then its label h is a morphism $h\colon M(B', \mathcal{X}', \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$ (in the opposite direction of the arc) which is specified by a mapping $h\colon \Delta \to B^*$ where $\Delta \subseteq B'$ is some subset (possibly empty) of constants with $\Delta \cap A = \emptyset$. We assume that such a map h extends to a $A \cup \mathcal{X}'$-morphism $h\colon M(B', \mathcal{X}', \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$ by leaving all letters in $(B' \cup \mathcal{X}') \setminus \left\{f(d) \;\middle|\; d \in \Delta \cup \overline{\Delta}, f \in H\right\}$ invariant. Since $h(\Delta) \subseteq B'^*$, the restriction of h also defines a morphism $h\colon M(B', \theta', \mu') \to M(B, \theta, \mu)$. Note that we use the same letter h for both morphisms. There will be no risk of confusion.

Since $B' \subseteq C$, the morphism h also induces an endomorphism of C^* which respects the involution assuming $h(c) = c$ for all $c \in C \setminus B'$. However, outside B' neither the action of H nor the value of μ is defined, so C^* is not an H-N-monoid. It is simply a free monoid with involution, and we can read the label always as an endomorphism of the free monoid with involution C^*.

New constants appear only by compression. If a word w is replaced a letter c by specifying $h(c) = w$, then we will automatically set $\mu(c) = \mu(w)$, $h(\overline{c}) = \overline{w}$, $h(f(c)) = f(h(c))$, and hence: $f(c) = c \iff f(w) = w$ for all $f \in H$. By definition of N, the second component in $\mu(w)$ is a word $u \in A^*$ of length at most $\log |H|$

such that the stabilizer H_w satisfies $H_c = H_w = H_u$. In particular, we have an efficient test whether $f(c) = c$ for all $f \in H$: we just check that $f(a) = a$ for all letters a which appear in the word u. The crucial observation is that whenever

$$(W_s, B_s, \mathcal{X}_s, \theta_s, \mu_s) \xrightarrow{h_{s+1}} \cdots \xrightarrow{h_t} (W_t, B_t, \mathcal{X}_t, \theta_t, \mu_t)$$

is a labeled path and $w \in B_t^*$ is word, then $h = h_{s+1} \cdots h_t$ can be viewed either as a morphism $h\colon M(B_t, \theta_t, \mu_t) \to M(B_s, \theta_s, \mu_s)$ or as an endomorphism of C^*. If we have $w \in B_t^*$, then h defines a word $h(w) \in B_s^*$ and the corresponding element $h(w) \in M(B_s, \theta_s, \mu_s)$. By ε we denote the identity endomorphism on C^*. Then ε appears as the label of transitions $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (W', B', \mathcal{X}', \theta', \mu')$ where $h\colon M(B', \theta', \mu') \to M(B, \theta, \mu)$ is a morphism with $h(a) = a$ for all $a \in B'$. For example, the label ε might appear when $B' \subseteq B$ or $\theta' \subseteq \theta$, etc.

7.1. **States.** We define the states of the $\mathcal{T}$ as the set of extended equations according to Definition 5.4. Thus, every state E is of the form $E = (W, B, \mathcal{X}, \theta, \mu)$.

7.1.1. *Initial state.* The *initial state* is $E_{\text{init}} = (W_{\text{init}}, A, \mathcal{X}, \emptyset, \mu_0)$.

7.1.2. *Final states.* A state $(W, B, \emptyset, \emptyset, \mu)$ is *final* if

(1) $W = \overline{W}$ and uses no variables.
(2) The word W has a prefix of the form $\# d_1 \# \cdots \# d_k \#$ where d_i are the *distinguished* letters mentioned in Theorem 4.3.

7.2. **Transitions.** We denote a transition as as $E \xrightarrow{h} E'$ and for both kinds, substitutions and compressions, we put some additional length restrictions on h. For example, we allow $h(c) = 1$ for a letter c only if E' is a final state. Thus labels on paths not ending in a final state are never length decreasing morphisms. Moreover, we require that $h(c)$ is not too long. If h is specified by a set Δ', then we require $\sum_{c \in \Delta'} |h(c)| < |W|$ where $E = (W, B, \mathcal{X}, \theta, \mu)$. These length restrictions are not used in the proof of the soundness result Proposition 7.5. We need them when proving completeness for a finite subautomaton of $\mathcal{T}$.

7.2.1. *Substitution transitions.*

Definition 7.1. *A substitution transition is denoted as*

$$(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (\tau(W), B', \mathcal{X}', \theta', \mu').$$

We must have $B \subseteq B'$ and we require that the transition is defined by a B-morphism $\tau\colon M(B, \mathcal{X}, \theta, \mu) \to M(B', \mathcal{X}', \theta', \mu')$ and a B-morphism $h\colon M(B', \mathcal{X}', \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$ such that $|h(b)| = 1$ for all $b \in B'$. In particular, h is length preserving.

In the case that some variable is typed in the source node $(W, B, \mathcal{X}, \theta, \mu)$, that is $\theta \neq \emptyset$, then we add the following restrictions:

- *$\mathcal{X}' \subseteq \mathcal{X}$. (Thus, for $\theta \neq \emptyset$ the set of variables cannot increase.)*
- *If $X \in \mathcal{X}'$, then $\theta(X) = \theta'(X)$. In particular, $\theta(X)$ is defined if and only if $\theta'(X)$ is defined.*
- *If $\theta(X)$ is not defined, then $\tau(X) = X$.*
- *If $\theta(X)$ is defined, then $\tau(X) \in \theta(X)^* \cup \theta(X)^* \, X \, \theta(X)^*$.*

We say that a substitution transition is *special* if $B = B'$. This implies that the label h is is the identity on $M(B, \theta, \mu)$; and therefore the label will be $h = \varepsilon = \mathrm{id}_{C^*}$. Later it would be enough to only consider special substitution transitions. However this would not simplify the following proof.

Lemma 7.2. *Let $E = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (\tau(W), B', \mathcal{X}', \theta', \mu') = E'$ be a substitution transition. If σ' solves E' then $\sigma = h\sigma'\tau$ solves E. In particular, if $\alpha\colon M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ is an A-morphism, then (α, σ) and $(\alpha h, \sigma')$ are entire solutions with $\alpha\sigma(W) = \alpha h\sigma'(W')$.*

Proof. Recall by Definition 5.5 to prove $\sigma = h\sigma'\tau$ solves E we must show two things: $\sigma(W) = \sigma(\overline{W})$ and whenever X is typed and $p = \theta(X)$ we must have $\sigma(X) \in p^*$.

We begin by checking that $\theta \neq \emptyset$ implies $h\sigma'\tau(X) \in \theta(X)^*$ for all typed variables. Consider a typed variable $X \in \mathcal{X}$. The first case is: $\tau(X) \in \theta(X)^* X \theta(X)^*$. Hence $X \in \mathcal{X}'$. By definition, $\theta'(X)$ is defined and $\theta(X) = \theta'(X) = p \in B^*$. Then $\tau(X) \in p^* X p^*$, too. Hence, $\sigma'\tau(X) \in p^*$ because every solution σ' has to satisfy $\sigma'(X) \in p^*$. Since $p \in B^*$ and h is a B-morphism we have $h(p) = p$. Therefore $h\sigma'\tau(X) \in \theta(X)^*$ in the first case. The second case is $\tau(X) \in p^*$ where $p = \theta(X) \in B^*$. Again, we can conclude $h\sigma'\tau(X) \in p^*$. Thus, in both cases: whenever $X \in \mathcal{X}$ is typed, then $h\sigma'\tau(X) \in \theta(X)^*$.

Since h, σ', and τ are B-morphisms, so is their composition $h\sigma'\tau$. Since σ' is a solution of $W' = \tau(W)$, we have $\sigma'(W') = \sigma'(\overline{W'})$. Hence, $\sigma'\tau(W) = \sigma'\tau(\overline{W})$ since $\overline{\tau(W)} = \tau(\overline{W})$ because τ respects the involution. It follows that $h\sigma'\tau(W) = h\sigma'\tau(\overline{W})$, so $\sigma(W) = \sigma(\overline{W})$. Thus, $h\sigma'\tau$ is a solution at E. As a consequence, $(\alpha, h\sigma'\tau)$ and $(\alpha h, \sigma')$ are both entire solutions because h is a B-morphism and $A \subseteq B$.

$$\begin{tikzcd}
M(B, \mathcal{X}, \theta, \mu) \arrow[r, "\tau"] \arrow[d, "\sigma"] & M(B', \mathcal{X}', \theta', \mu') \arrow[d, "\sigma'"] \\
M(B, \theta, \mu) & M(B', \theta', \mu') \arrow[l, "h"']
\end{tikzcd}$$

□

7.2.2. *Compression transitions.* Compressions are defined only if $\mathcal{X} = \mathcal{X}'$. They leave the variables invariant, but we encounter both situations $B \subseteq B'$ or $B' \subseteq B$. However, in case that $\theta \neq \emptyset$ the situation is more subtle than for substitutions, and we need again technical restrictions in order to guarantee soundness.

Definition 7.3. *A* compression transition

$$(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu')$$

is defined in $\mathcal{T}$ if $h\colon M(B', \mathcal{X}, \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$ is an $(A \cup \mathcal{X})$-morphism such that the following conditions hold.

- *We have $W = h(W')$*
- *$h(b')$ can be written as a word in B^* for every $b' \in B'$ and $|h(c)| \geqslant 1$ for all $c \in B'$ unless $E' = (W', B', \mathcal{X}, \theta', \mu')$ is a final state.*
- *h is specified by a mapping $h\colon \Delta' \to B^*$ with $\Delta' \subseteq B'$ such that*

$$\sum_{c \in \Delta'} |h(c)| < |W|.$$

- *A variable X is typed using θ' if and only if it is typed using θ.*
- *There is some $e \geqslant 1$ such that for all typed variables we have*

$$h(\theta'(X)) = \theta(X)^e.$$

Note that for a given $(A \cup \mathcal{X})$-morphism $h\colon M(B', \mathcal{X}, \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$ the conditions to be a compression transition are effective.

Lemma 7.4. *Let $E = (h(W'), B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu') = E'$ be a compression and σ' be a solution at E'. Then there exists a B-morphism*

$$\sigma\colon M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$$

such that $h\sigma'(X) = \sigma(X)$ for all $X \in \mathcal{X}$. The B-morphism σ satisfies the following conditions. If $\alpha\colon M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ is an A-morphism, then first, (α, σ) is an entire solution at E and second, $(\alpha h, \sigma')$ is an entire solution at E'. Moreover, $\alpha\sigma(W) = \alpha h\sigma'(W')$.

Proof. Define $\sigma(X) = h\sigma'(X)$ for all variables and $\sigma(b) = b$ for all $b \in B$. This defines a B-morphism $\sigma\colon M(B, \mathcal{X}, \emptyset, \mu) \to M(B, \theta, \mu)$ since $M(B, \mathcal{X}, \emptyset, \mu)$ is a free monoid. (There is no type yet on the left.) Let us show first that $(x, y) \in \theta$ implies $\sigma(x) = \sigma(y)$ in the monoid $M(B, \mathcal{X}, \theta, \mu)$. That is, σ induces a B-morphism (which we also denote by σ) $\sigma\colon M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$.

For $(x, y) \in \theta$ with $x, y \in B^*$ the assertion $\sigma(x) = \sigma(y)$ is trivial because σ leaves B^* invariant. Thus, it is enough to consider a defining relation of the form $(Xp, pX) \in \theta$ where $p = \theta(X) \in B^*$. Because of Definition 7.3 we know that X is typed on the right hand side $M(B', \mathcal{X}, \theta', \mu')$, too. Let $q = \theta'(X) \in B'^*$. Thus, $h(q) = p^e$ for some $e \geqslant 1$ according to the last condition in Definition 7.3. Since $\sigma'(X) = q^\ell$ for some $\ell \geqslant 0$, we conclude $h\sigma'(X) = p^{e\ell}$. Hence, whenever $(Xq, qX) \in \theta'$, then $h\sigma'(Xp) = p^{1+e\ell} = h\sigma'(pX)$ in $M(B, \theta, \mu)$ since $(Xp, pX) \in \theta$.

So far we have shown that σ is a well-defined morphism such that $\sigma(X) = h\sigma'(X)$. This implies $\sigma(h(X)) = h\sigma'(X)$ for all variables. For a constant $b \in B$ we have $\sigma(h(b)) = h(b) = h(\sigma'(b))$. Hence $\sigma h = h\sigma'$ and this means that the diagram in Figure 4 commutes. The morphism $h\colon M(B', \theta', \mu') \to M(B, \theta, \mu)$ in Figure 4 denotes the restriction of the morphism $h\colon M(B', \mathcal{X}', \theta', \mu') \to M(B, \mathcal{X}, \theta, \mu)$, too. Let $W = h(W')$ and hence, $\overline{W} = h(\overline{W'})$. In order to see that (α, σ) is an entire solution at E we use $\sigma h = h\sigma'$ and we content ourselves to consider the following line of equations:

$$\begin{tikzcd}
M(B, \mathcal{X}, \theta, \mu) \arrow[d, "\sigma"] & M(B', \mathcal{X}, \theta', \mu') \arrow[l, "h"'] \arrow[d, "\sigma'"] \\
M(B, \theta, \mu) & M(B', \theta', \mu') \arrow[l, "h"']
\end{tikzcd}$$

Figure 4. $h\sigma = \sigma' h$.

$$\sigma(W) = \sigma h(W') = h\sigma'(W') = h\sigma'(\overline{W'}) = \sigma h(\overline{W'}) = \sigma(\overline{W}).$$

In particular, $\alpha\sigma(W) = \alpha\sigma h(W') = \alpha h\sigma'(W')$. It is also clear that $(\alpha h, \sigma')$ is an entire solution at E' since h leaves A invariant. □

Proposition 7.5. *Let $E_0 \xrightarrow{h_1} \cdots \xrightarrow{h_t} E_t$ be a path in $\mathcal{T}$ of length t, where $E_0 = (W_{\text{init}}, A, \mathcal{X}, \emptyset, \mu_0)$ is an initial and $E_t = (W, B, \emptyset, \emptyset, \mu)$ is a final state. Then E_0 has an entire solution $(\mathrm{id}_{A^*}, \sigma)$ with $\sigma(W_{\text{init}}) = h_1 \cdots h_t(W)$. In particular, for $X \in \mathcal{X}$ we have $\sigma(X) = h_1 \cdots h_t(d_X)$; and $\mathcal{T}$ is sound in the sense of (15).*

Proof. Since E_t is final, it has a unique solution $\sigma_t = \mathrm{id}_{B^*}$. By the lemmas above, we obtain a solution σ at E_0 such that $\mathrm{id}_{A^*}\sigma(W_{\text{init}}) = \mathrm{id}_{A^*} h_1 \cdots h_t \mathrm{id}_{B^*}(W)$. Hence, $(\mathrm{id}_{A^*}, \sigma)$ is an entire solution as desired. □

8. The intermediate automaton $\mathcal{F}$

Proposition 7.5 states that the large automaton $\mathcal{T}$ is sound. This property cannot be destroyed by removing states or transitions. That is, every subautomaton of $\mathcal{T}$ is sound, too. We define a subautomaton $\mathcal{F}$ of $\mathcal{T}$ as follows. All extended equations E are states of $\mathcal{F}$, so the state set is the same infinite set as for $\mathcal{T}$. However, for transitions we are more restrictive. To define transitions, let us first define a *weight* for equations and states. The definition is tailored that all compression transitions and certain substitution transitions reduce the weight of the state.

Definition 8.1. *Let $E = (W, B, \mathcal{X}, \theta, \mu)$ be an extended equation where (as usual) $W \in (A \cup \mathcal{Y})^*$ is represented as a word. The* weight *of the equation $\|W\|$ is defined by*

$$\|W\| = \begin{cases} |W| + 30\delta \sum_{Y \in \mathcal{Y}} |W|_Y & \text{if } E \text{ is not final,} \\ 0 & \text{otherwise, that is: } E \text{ is final.} \end{cases} \tag{16}$$

The weight *of the state $\|E\|$ is a pair of natural numbers $\|E\| = (\|W\|, |B|)$.*

For $\ell \in \mathbb{N}$ we order tuples in $\mathbb{N}^\ell$ lexicographically. For example $(0, 42) < (1, 0)$, but $(1, 0, 42) > (0, 10, 100)$; and we use the fact that there are no infinite descending chains in $\mathbb{N}^\ell$. Consider any transition $E \xrightarrow{h} E'$ in $\mathcal{T}$. Then we always have $|E| < |E'|$ unless the transition is a substitution transition where at least one variable that appears in W pops out a constant.

Remark 8.2. *The definition of $\|W\|$ is invariant under the word representation of W. This follows because $\sum_{Y \in \mathcal{Y}} |x|_Y = \sum_{Y \in \mathcal{Y}} |y|_Y$ for all $(x, y) \in \theta$. Second, the advantage to use the weight $\|W\|$ in $\|E\|$ (instead of using the more straightforward choice of $(|W|, |B|)$ for $\|E\|$) is that following a substitution transition, which does nothing but replace variables X by $\sigma(X)$ for $|\sigma(X)| \leqslant 30\delta$, leads to a state of smaller weight.*

A transition $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$ in $\mathcal{T}$ belongs to $\mathcal{F}$ if and only if the following properties are satisfied.

- If $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$ is a substitution transition, then $W' = \tau(W)$, $B = B'$, and $h = \varepsilon$.
- If $E = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu') = E'$ is a compression transition, then $W = h(W')$ and $\|E'\| < \|E\|$.

The focus for the remaining part of the proof is on completeness. A subautomaton $\mathcal{A}$ of $\mathcal{T}$ is called *complete* if it holds:

$$\mathrm{Sol}(\mathcal{S}) \subseteq \{(h(d_1), \dots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A})\}. \tag{17}$$

Since every subautomaton of $\mathcal{T}$ is also sound in the sense (15) we see that every complete subautomaton is sound and complete.

Proposition 8.3. *Let $\mathcal{A}$ be a trim, finite subautomaton of $\mathcal{F}$. If $\mathcal{A} \neq \emptyset$, then $\mathcal{S}$ has at least one solution. If $\mathcal{A}$ contains a directed cycle, then $\mathcal{S}$ has infinitely many solutions. Moreover, if $\mathcal{A}$ is complete, then the converse of both assertions is true.*

Proof. If $\mathcal{A} \neq \emptyset$, then $\mathcal{S}$ has at least one solution by Proposition 7.5. Now assume that $\mathcal{A}$ contains a directed cycle. By hypothesis $\mathcal{A}$ is trim. Hence, there is an accepting path with a directed cycle and this cycle doesn't involve any final state as final states are without outgoing arcs. Let $E_s \xrightarrow{h_s} \cdots \xrightarrow{h_t} E_t = E_s$ be this cycle. Without restriction we have $t > s$ and $\|E_s\| = \|E_{s+1}\|$ because $\mathbb{N}^\ell$ admits no infinite strictly descending chains. This means $E_s \xrightarrow{\varepsilon} E_{s+1}$ must be a substitution transition which is defined by some τ with $|\tau(X)|_a \geqslant 1$ for some X where X appears in the equation belonging to E_s and a is a constant. Hence, on some accepting path we can pop out an arbitrary number of letters of X. Since on paths from an initial state to E_s the labels are non-erasing endomorphisms, we see that we can make $\sigma(X) \in A^*$ at the initial state E_{init} larger and larger. Thus, there are infinitely many solutions. The converse, under the assumption that (17) holds, is trivial. □

9. Towards completeness

During the completeness proof we always work with a state $E = (W, B, \mathcal{X}, \theta, \mu)$ and a given entire solution (α, σ). Starting at a triple (E, α, σ) where E is a standard state, we describe a deterministic process which yields a path $(h_1, \ldots, h_t)$ inside the (infinite) automaton $\mathcal{F}$ from E to some final state $E_t = (W_t, B_t, \emptyset, \emptyset, \mu_t)$ so that $\alpha\sigma(W) = h_1 \cdots h_t(W_t)$. Thus, $\mathcal{F}$ is complete. The crucial property is that we are able to control the lengths of all intermediate equations W_i for $1 \leqslant s \leqslant t$ by

$$|W_s| \leqslant |W_s| + \mathcal{O}(\delta n). \tag{18}$$

We make sure that whenever we see an intermediate state E_s where $\theta_s \neq \emptyset$, then θ_s has a special structure. Moreover, when we follow a compression transition then we make sure the soundness condition holds for the corresponding label according Definition 7.3.

We then can deduce Theorem 4.3 because for defining the complete NFA $\mathcal{A}_\mathcal{S}$ mentioned in the theorem it is enough to consider the starting point

$$E = E_{\text{init}} = (W_{\text{init}}, A, \mathcal{X}, \emptyset, \mu_0).$$

Since $|W_{\text{init}}| \leqslant n$ we can ensure $\mathcal{A}_\mathcal{S}$ is finite by allowing extended equations in $\mathcal{A}_\mathcal{S}$ only if the corresponding equation satisfies a concrete length bound in $\mathcal{O}(\delta n) = \mathcal{O}(|H|\, n^2)$. Moreover, we impose that $\mathcal{A}_\mathcal{S}$ is trim. We will come back later to these issues. For the moment we work in the infinite automaton $\mathcal{F}$ and there is no length bound for the equation W.

9.1. **Dummy variables denoting the empty word.**

In the following it is convenient to have the following notation at our disposal. We introduce purely formal symbols of the form (f, D) where $f \in H$ and D is called *dummy variable*, but the symbol (f, D) is just another explicit notation for the empty word 1. The dummy variable D is never listed in $\mathcal{Y}$. Its only purpose is that we have a unified notation for local equations (and avoid case distinctions). Since $(f, D) = 1$ every morphism maps (f, D) to 1. The advantage is that with the help of a dummy variable, we may, whenever convenient, assume that every local equation has the form

$$u(f, X)w(g, Y)v = uZv.$$

Here X, Y, Z are (perhaps dummy) variables and u, v, w are words over constants.

9.2. **The weight of entire solution and the forward property.**

We need a termination condition for the following compression procedure. Therefore we define a *weight* $\|E, \alpha, \sigma\| \in \mathbb{N}^3$ for the triple (E, α, σ) where $E = (W, B, \mathcal{X}, \theta, \mu)$ is a state with an entire solution (α, σ) by

$$\|E, \alpha, \sigma\| = \begin{cases} \left(\sum_{X \in \mathcal{X}} |\alpha\sigma(X)|, \|W\|, |B|\right) & \text{if } E \text{ is not final,} \\ (0, 0, 0) & \text{otherwise.} \end{cases}$$

At non-final states the weights $\|W\| = |W| + 30\delta \sum_{Y \in \mathcal{Y}} |W|_Y$ and $\|E\| = (\|E\|, |B|)$ were defined in Equation (16). Thus, actually for all states, we can write $\|E, \alpha, \sigma\|$ as a pair

$$\|E, \alpha, \sigma\| = (\sum_{X \in \mathcal{X}} |\alpha\sigma(X)|, \|E\|).$$

Being at (E, α, σ) we say that a transition $E \xrightarrow{h} E' = (W', B', \mathcal{X}', \theta', \mu')$ satisfies the *forward property* if E' has an entire solution $(\alpha h, \sigma')$ such that first $\|E', \alpha h, \sigma'\| < \|E, \alpha, \sigma\|$ and second, $\alpha h \sigma'(W') = \alpha\sigma(W)$.

Following a transition $E \xrightarrow{h} E'$ which satisfies the forward property means that we switch from (E, α, σ) to $(E', \alpha h, \sigma') = (E', \alpha', \sigma')$. Typically after each such step we rename the tuple $(W', B', \mathcal{X}', \theta', \mu', \alpha', \sigma')$ as $(W, B, \mathcal{X}, \theta, \mu, \alpha, \sigma)$. Since using a transition satisfying the forward property reduces the weight $\|E, \alpha, \sigma\|$, there are no infinite paths of transitions where all transitions on the path satisfy the forward property.

9.3. **Meta rules.** Let $E = (W, B, \mathcal{X}, \theta, \mu)$ be a state with an entire solution (α, σ). We apply the following meta rules whenever possible.

9.3.1. *Remove variables with short solutions.* If $|\sigma(Y)| \leqslant 30\delta$ for some variable $Y \in \mathcal{Y}$ such that $|W|_Y \geqslant 1$, then follow a substitution transition $E \xrightarrow{\varepsilon} E'$ which is defined by a B-morphism τ such that $\tau(Z) = \sigma(Z)$ if $|\sigma(Z)| \leqslant 30\delta$ and $\tau(Z) = Z$ otherwise. The state $E' = (\tau(W), B, \mathcal{X}', \theta', \mu')$ uses the same set of constants, but we have $\mathcal{X}' \subsetneq \mathcal{X}$. We also have $\theta' \subseteq \theta$ and μ' is the restriction of μ. (Thus, according to our convention we can also write $E' = (\tau(W), B, \mathcal{X}', \theta, \mu)$.) Let σ' be the restriction of σ, then (α, σ') is an entire solution at E' and we have $\alpha\varepsilon\sigma(W) = \alpha\sigma'\tau(W)$. Moreover, $\|W'\| < \|W\|$. Hence, $\|E'\| < \|E\|$ and the transition reduces the weight of the state.

As a consequence, whenever we are at a state $E = (W, B, \mathcal{X}, \theta, \mu)$ with an entire solution (α, σ), then we assume $|\sigma(Y)| > 30\delta$ for all $Y \in \mathcal{Y}$ where $|W|_Y \geqslant 1$.

9.3.2. *Remove useless constants.* We say that a letter $a \in B \setminus A$ is *useless* (with respect to σ) if $|\sigma(W)|_{f \cdot a} = 0$ for all $f \in H$. Note that a letter $a \in A$ is never useless. If B contains a useless letter a, then define $B' = B \setminus \{f \cdot a \mid f \in H\}$. The inclusion of B' into B defines canonical embeddings $M(B', \mathcal{X}, \theta, \mu) \to M(B, \mathcal{X}, \theta, \mu)$ and $M(B', \theta, \mu) \to M(B, \theta, \mu)$ such that $W \in M(B', \mathcal{X}, \theta, \mu)$ and $\sigma(W) \in M(B', \theta, \mu)$. The state $E' = (W, B', \mathcal{X}, \theta, \mu)$ has an entire solution (α, σ') where σ' is the restriction of σ. Moreover, $\|E'\| < \|E\|$. Hence, we can follow the compression transition $E \xrightarrow{\varepsilon} (W, B', \mathcal{X}, \theta, \mu)$ which satisfies the forward property.

As a consequence, whenever we are at a state $E = (W, B, \mathcal{X}, \theta, \mu)$ with an entire solution (α, σ), then we assume that B doesn't contain any useless letters.

Remark 9.1. *We may have that $B \setminus A$ contains letters that are not H-visible, but a solution σ uses them. Removing useless letters does not remove such letters.*

9.3.3. *Moving to a final state.* Let $E = (W, B, \emptyset, \emptyset, \mu)$ be a standard state without any variables and with an entire solution (α, σ). Then $\sigma = \mathrm{id}_B$ is the identity on $M(B, \emptyset, \emptyset, \mu) = M(B, \emptyset, \mu)$ and we have $W = \overline{W}$. If E is final, there is nothing to do. Hence, we assume that E is not final. Since $W = \overline{W}$, Definition 5.4 tells us

$$W = \# x_1 \# \cdots \# x_k \# u \# \# \overline{u} \# \overline{x_k} \# \cdots \# \overline{x_1} \#.$$

Hence we can enlarge B to a set B' which contains all distinguished d_i for $1 \leqslant i \leqslant k$. (By our convention none of the d_i belongs to B because the state E is not final.) We define a B-morphism $h \colon M(B', \emptyset, \mu) \to M(B, \emptyset, \mu)$ by letting $h(d_i) = x_i$. Moreover, we let

$$W' = \# d_1 \# \cdots \# d_k \# u \# \# \overline{u} \# \overline{d_k} \# \cdots \# \overline{d_1} \#.$$

We have no variables and $W' = \overline{W'}$. Hence $E' = (W', B', \emptyset, \emptyset, \mu')$ is final. The entire solution at E' is $(\alpha h, \mathrm{id}_B)$ and we have $\alpha h(W') = \alpha(W)$ since none of the d_i belong to B. Since $\|E, \alpha, \mathrm{id}_B\| > (0, 0, 0) = \|E', \alpha h, \mathrm{id}_B\|$ the compression transition $E \xrightarrow{h} E'$ satisfies the forward property. Hence, we are done.

As a consequence, whenever we are at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ with an entire solution (α, σ), then we assume that $\mathcal{X} \neq \emptyset$. Moreover according to the other meta rules we have $|\sigma(X)| \geqslant 30\delta$ for all $X \in \mathcal{X}$ and every constant $b \in B$ is H-visible in W.

10. Compression round: the first phase

We perform the compression in rounds. Each round has two phases. The first phase is called *δ-periodic compression*, the second one is called *pair compression.* During δ-periodic compression we perform all meta rules whenever possible. Recall how meta rules decrease the weight $(\|W\|, |B|)$ at states: removing a variable makes $\|W\|$ smaller, removing useless letters doesn't change $\|W\|$, but it makes B smaller. Moving to a final state decreases the weight of the state down to $(0, 0)$ (which was the exceptional weight at final states) . None of these rules increases the sum $\sum_{X \in \mathcal{X}} |\alpha\sigma(X)|$. Therefore all meta rules satisfy the forward property according to Section 9.2.

10.1. A simple, but useful, estimation.

During the rounds the length oscillates but it can be bounded by some function in $\mathcal{O}(\delta n)$. In order to obtain such a bound we will later apply the following fact twice with different parameters.

Lemma 10.1. *Let $0 \leqslant q < 1$ and $c \geqslant 1$ for some real constants q, c, and let $s \colon \mathbb{N} \to \mathbb{N}$ be a function with $s(0) \leqslant \frac{c}{1-q}\delta n$ and which satisfies a bound*

$$s(t+1) \leqslant q\, s(t) + c\delta n$$

for all $t \in \mathbb{N}$. Then $s(t) \leqslant \frac{c}{1-q} \cdot \delta n$ for all $t \in \mathbb{N}$.

Proof. The statement is true for $t = 0$. Assuming it is true for $t \geqslant 0$ then

$$s(t+1) \leqslant qs(t) + c\delta n \leqslant q\frac{c}{1-q}\delta n + c\delta n = \frac{c}{1-q} \cdot \delta n. \tag{19}$$

□

In $\mathcal{O}$-notation (19) reads as: if $s(0) \geqslant 0$ and $s(t+1) \in qs(t) + \mathcal{O}(\delta n)$, then $s(t) \in \mathcal{O}(\delta n)$ for all t.

10.2. Alphabet reduction at standard states.

During our procedures we introduce more and more letters, so the set B grows, and removing useless letters is not enough to keep the size of B in $\mathcal{O}(|H| \cdot |W|)$.

The following procedure which we call *alphabet reduction* is not a meta rule (which we may apply whenever possible). If we call the procedure we explicitly say so. When we call it we wish that $B \setminus A$ contains only H-visible letters in W.

We begin at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ with an entire solution (α, σ) where there is some letter $b \in B$ which is not H-visible. Hence $|W|_{f \cdot b} = 0$ for all $f \in H$. Removing useless letters is a meta rule. Hence, we may assume without restriction that all letters are useful and therefore we may assume $|\sigma(X)|_b \geqslant 1$ for some variable. (That is, we are in the situation of Remark 9.1.) Define

$$B' = A \cup \left\{ a \in B \;\middle|\; \exists f \in H : |W|_{f(a)} \geqslant 1 \right\}.$$

Then we have $W \in M(B', \mathcal{X}, \emptyset, \mu)$. The procedure will takes us (via a compression transition defined by the inclusion $B' \subseteq B$) to the state $E' = (W, B', \mathcal{X}, \emptyset, \mu)$. Since $b \in B \setminus B'$ we have $|B'| < |B|$ and therefore $\|E'\| < \|E\|$, too.

It is here where the notion of entire solution becomes important. We have $\alpha \colon M(B, \emptyset, \mu) \to M(A, \emptyset, \mu_0)$, so we can define a B'-morphism $\beta \colon M(B, \emptyset, \mu) \to M(A, \emptyset, \mu_0)$ by $\beta(b) = \alpha(b)$ for $b \in B \setminus B'$. Since $M(B, \emptyset, \mu) = B^*$ is a free monoid, we don't have to worry to check defining relations. Moreover, $\sigma' = \beta\sigma$ is solution at $E' = (W, B', \mathcal{X}, \emptyset, \mu')$. Thus, we can switch from (E, α, σ) to $(E, \alpha, \sigma') = (E, \alpha, \beta\sigma)$

via the compression transition $(W, B, \mathcal{X}, \emptyset, \mu) \xrightarrow{\varepsilon} (W, B', \mathcal{X}, \emptyset, \mu)$. Since α is an A-morphism we obtain $\alpha = \alpha\beta$. Hence, $\alpha\sigma(W) = \alpha\beta\sigma(W) = \alpha\varepsilon\sigma'(W)$ as desired.

As a consequence, whenever we perform an alphabet reduction, then we arrive at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ with an entire solution (α, σ) such that every letter in $B \setminus A$ is H-visible in W. This means that after alphabet reduction the size of B is at most $|H| \cdot (|A| + |W|)$.

10.3. **Mapping the positions from $\sigma(W)$ to W.** Let $E = (W, B, \mathcal{X}, \emptyset, \mu)$ be a state with an empty type θ and let $\sigma \colon M(B, \mathcal{X}, \emptyset, \mu) \to M(B, \emptyset, \mu)$ be any B-morphism. Recall that $\{1, \ldots, m\}$ (resp. $\{1, \ldots, \ell\}$) denotes the set of positions of $\sigma(W)$ (resp. W). Then σ induces a mapping π_σ from $\{1, \ldots, m\}$ to $\{1, \ldots, \ell\}$ as follows. We define π_σ from left-to-right. We let $\pi_\sigma(1) = 1$. The first position in $\sigma(W)$ is labeled with # and so is the first position in W. No other position than 1 is mapped to 1. We shall keep the invariant that $\sigma(W[1, m']) = W[1, \ell']$ if m' is the largest position which is mapped to ℓ'. In particular, we have $\sigma(W[m'+1, m]) = W[\ell'+1, \ell]$.

Now assume $\pi_\sigma(i)$ is already defined for all $1 \leqslant i \leqslant m'$ and $m' \leqslant m$. If $m' = m$ we are done. Otherwise we have $m' < m$ and we consider $\pi_\sigma(m') = \ell'$. By the invariant we know $\ell' < \ell$. We look at the label of the position $\ell'+1$. It is labeled by a letter in W and there are two cases. In the first case the label is a constant $b \in B$. In this case we let $\pi_\sigma(m'+1) = \ell'+1$. In the second case the label is of the form Y with $Y \in \mathcal{Y}$. In that case we map all positions in the interval $[m'+1, m'+|\sigma(Y)|\,]$ to the single position $\ell'+1$.

Note that $\pi_\sigma \colon \{1, \ldots, m\} \to \{1, \ldots, \ell\}$ enjoys the following properties. If ℓ is a position of W which is labeled by a constant $b \in B$, then ${\pi_\sigma}^{-1}(\ell)$ is a single position in $\sigma(W)$ which is labeled by b, too. If ℓ is a position of W which is labeled by a variable $Y \in \mathcal{Y}$, then ${\pi_\sigma}^{-1}(\ell)$ is a interval of length $|\sigma(Y)|$ in $\sigma(W)$. The label of that interval is just $\sigma(Y)$.

Definition 10.2. *We say that a position m' of $\sigma(W)$ is* visible (in W) *if $\pi_\sigma(m')$ is a constant. Otherwise it is called* invisible. *An interval $[i, j]$ of positions of $\sigma(W)$ is* visible (in W) *(resp.* invisible*) if all positions in that interval are visible (resp. invisible) positions. If $[i, j]$ contains an invisible position, but $|\pi_\sigma[i, j]| \geqslant 2$, then we say that the interval $[i, j]$ is* crossing.

10.4. **The start of a compression round.** Each compression round starts at a standard state $E_r = (W_r, B_r, \mathcal{X}_r, \emptyset, \mu_r)$ with an entire solution (α_r, σ_r). We may assume that no meta rule is applicable. The very first step is now an alphabet reduction. For simplicity, we denote the state again by $E_r = (W_r, B_r, \mathcal{X}_r, \emptyset, \mu_r)$ and we have $|B_r| \leqslant |H| \cdot |W_r|$.

10.5. **δ-periodic compression.** For convenience we rename the tuple

$$(E_r, W_r, B_r, \mathcal{X}_r, \mu_r, \alpha_r, \sigma_r) = (E, W, B, \mathcal{X}, \mu, \alpha, \sigma).$$

At this point we know that no meta rule applies to E and that $|B| \leqslant |H| \cdot |W|$.

Let us consider all very long maximal δ-periodic factors w of $\sigma(W)$ which have a maximal occurrence with at least one visible position. (By maximal occurrence we mean that w is not a factor of a longer δ-periodic word at that occurrence.) We assume that at least one such occurrence exists, otherwise we skip the main body of the δ-periodic compression and proceed directly to the end: Section 10.6.

We write $w = up^erv$ with $|u| = |v| = 3\delta$, p is primitive of length at most δ and r is a nonempty prefix of p. (Recall very long means $|w| \geqslant 10\delta$ so $|p^er| \geqslant 4\delta$.) By Lemma 6.4, we can encode the factor up^erv uniquely by writing the triple (p, r, e). Let us call u and v the *borders* of the very long maximal δ-periodic factor up^erv. Consider different occurrences up^erv and $u'p'^{e'}r'v'$ of very long maximal δ-periodic factors in $\sigma(W)$. If the occurrences overlap, then this overlap takes place in the borders only, because otherwise the occurrence of the factor was not maximal.

It follows that the number of occurrences very long maximal δ-periodic factors with at least one visible position is less than $|W|$. Thus we find some minimal index set Λ of size $|\Lambda| < |W|$ such that

$$F_\Lambda = \{u_\lambda p_\lambda^{e_\lambda} r_\lambda v_\lambda \mid \lambda \in \Lambda\}$$

is exactly the set of very long maximal δ-periodic factors of $\sigma(W)$ which have a maximal occurrence with at least one visible position.

The idea is that at the end we arrive at a state with a solution where all these occurrences are replaced by $u[r,s,\lambda]v$ where $[r,s,\lambda]$ is the notation for a fresh letter such that $p = rs$, $r \neq 1$, and $\lambda \in \Lambda$. We also *color* certain positions in W and $\sigma(W)$. At the end of the process a position will be *green* if and only if it is labeled by some new letter $[r,s,\lambda]$.

Note that λ is just a formal symbol: we need at most $\mathcal{O}(|W|)$ bits to encode it. We also define a set of primitive words

$$P_\Lambda = \{p_\lambda \mid u_\lambda p_\lambda^{e_\lambda} r_\lambda v_\lambda \in F_\Lambda\}.$$

We have

$$1 \leqslant |P_\Lambda| \leqslant |F_\Lambda| = |\Lambda|.$$

Next we consider fresh variables which are denoted as $[X, f(sr)]$ where $X \in \mathcal{X} \subseteq \mathcal{X}$, $f \in H$ and for certain $p_\lambda \in P_\Lambda$ and then for all $rs = p_\lambda$. These new variables will later be typed. We define the action of H by $g \cdot [X,p] = [X, g(p)]$ and the involution by $\overline{[X,p]} = [\overline{X}, \overline{p}]$. The idea is $\sigma([X,p]) \in p^*$; and thus, $\sigma(\overline{[X,p]}) \in \overline{p}^*$ and $\sigma((f,[X,p])) \in f(p)^*$. Note that $(f,[X,p]) = (g,[X,p])$ if and only if $g^{-1}f(p) = p$ and hence $g^{-1}f \in H_p$ is in the known stabilizer of p.

The following routine introduces these new variables using substitution transitions. Recall that defining $\tau(X) = w$ substitutes (f,X) by $f(w)$ and simultaneously $(f,\overline{X})$ by $f(\overline{w}) = \overline{f(w)}$ for all $f \in H$.

```
begin procedure (insert new variables)
Initialize a set of fresh variables by $\mathcal{X}_{\text{new}} = \emptyset$ and put $E = (W, B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \emptyset, \mu)$.
forall $X \in \mathcal{X}$ do
(Note this means we do the process once for $X$ and once for $\overline{X}$.)
    (1) Apply all meta rules whenever possible; in particular, $|\sigma(Y)| \geqslant 30\delta$ for all variables.
    (2) Let $q^d q'$ be the longest suffix of $\sigma(X)$ such that $q$ is primitive, $|q| \leqslant \delta$, and $q'$ is a prefix of $q$. If $|q^d q'| \leqslant 3\delta$, then do nothing.
    (3) If $|q^d q'| > 3\delta$, then define words $p$, $p'$, and $e \geqslant 0$ by $q^d q' = up^e p'$ with $|u| = 3\delta$, $|p| = |q|$, and $1 \neq p' \leqslant p$. (Note that $p$ is primitive: we have $p = q_2 q_1$ for some factorization $q = q_1 q_2$.) We enlarge $\mathcal{X}_{\text{new}}$ by a fresh variables $[X, sr]$ for all factorizations $p = rs$. Moreover, if we enlarge $\mathcal{X}_{\text{new}}$ by some $[X,p]$, then we also include $[X, f(p)]$ and $[X, f(\overline{p})]$ for all $f \in H$.
        We can write $\sigma(X) = xup^e p'$ with $|xu| \geqslant 3\delta$. Follow a substitution transition $E \xrightarrow{\varepsilon} E' = (\tau(W), B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \emptyset, \mu')$ which is defined by $\tau(X) = X[X,p]p'$ and define an entire solution at $E'$ by $(\alpha, \sigma')$ where $\sigma'(X) = xu$ and $\sigma'[X,p] = p^e$. The transition satisfies the forward property. (Due to the meta rules it can happen that $\mathcal{X}$ becomes smaller and/or that $\mathcal{X}_{\text{new}}$ is not enlarged at all.)
    (4) Rename $E', \tau(W), \mu', \sigma'$ as $E, W, \mu, \sigma$.
endforall
endprocedure
```

The number of new variables $[X,p] \in \mathcal{X}_{\text{new}}$ is bounded by $\mathcal{O}(|H|\,\delta n)$ because $|\mathcal{X}| \leqslant n$. The factor δ comes in because we consider all cyclic permutations sr of $p = rs$. The factor $|H|$ comes in because we close under H-action. On the other hand we don't need to list $[X, f(p)]$ in $\mathcal{X}_{\text{new}}$ for $1 \neq f \in H$ if $[X,p]$ is already listed. Thus, a list of $\mathcal{O}(\delta n)$ new variables suffices to specify the full set $\mathcal{X}_{\text{new}}$ (which is closed under the action of H and involution. Note that $\overline{[X,p]} = [\overline{X}, \overline{p}] \neq [X,p]$. So we keep the invariant that variables are not self-involuting.

From now on until the end of δ-periodic compression we only remove variables. So, if $\mathcal{Y}' = H \cdot \mathcal{X}'$ is the full set of variables we meet during the whole procedure, then we have:

$$|\mathcal{X}'| \in \mathcal{O}(\delta n). \tag{20}$$

Before the procedure we had $\mathcal{Y} = H \times \mathcal{X}$ and $\mathcal{X} \subseteq \mathcal{X}$, and $\sum_{Y \in \mathcal{Y}} |W|_Y \leqslant n$ as required for a standard state. The corresponding set after the procedure is $\mathcal{Y}_{\text{new}} = \mathcal{X}_{\text{new}} \cup H \times \mathcal{X}$. We have to check that $\sum_{Y \in \mathcal{Y}_{\text{new}}} |W|_Y \leqslant 3n$ (because otherwise it is not an extended equation as per Definition 5.4). This bound is immediate.

In the case $\mathcal{X}_{\text{new}} = \emptyset$ we are still at a standard state and $\sum_{Y \in \mathcal{Y}_{\text{new}}} |W|_Y \leqslant n$. For $\mathcal{X}_{\text{new}} \neq \emptyset$ we are not at a standard state because $\mathcal{Y}_{\text{new}}$ is not contained in $H \times \mathcal{X}$.

Since for $(W, B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \emptyset, \mu)$ the type θ is empty, we map the positions of $\sigma(W)$ to the positions W as explained above in Section 10.3. Consider any occurrence of a very long δ-periodic factor $w = up^e p'v$ in $\sigma(W)$ which is maximal and where at least one position is visible and where $|u| = |v| = 3\delta$. Consider the occurrence of all very long maximal δ-periodic factors w' where $w' \in \{g \cdot w, g \cdot \overline{w} \mid g \in H\}$. Each factor w' can be written as $w' = u'w''v'$ where $|u'| = |v'| = 3\delta$.

For all maximal occurrences of these factors w' let us color the inner positions belonging to w'' green. Then only green positions are mapped to a variable $[X,q] \in \mathcal{X}_{\text{new}}$. It is also clear that we we can write $q = sr$ for some factorization $p = rs$. Let us transport the green color to the corresponding positions in W. Then for all positions in W which are labeled by a variable it holds that the position is green if and only if it is new variable. Note that green positions in $\sigma(W)$ are separated by words of length at least 3δ.

In the next procedure we will introduce a type θ which consists of defining relations of the form $[X, aq]a = a[X, qa]$, but it will be enough to apply such a rule where both positions in W are green. Hence, the color of the positions will not be altered under this restriction. In order to define θ we use $\mathcal{X}_{\text{new}} \neq \emptyset$, otherwise we skip the next procedure.

begin procedure (introduce a type θ)

1. Define the type θ by $\{([X, as]a,\ a[X, sa]) \mid a \in B \wedge [X, as] \in \mathcal{X}_{\text{new}}\} \cup \{([X,p]p,\ p[X,p]) \mid [X,p] \in \mathcal{X}_{\text{new}}\}$ Note that $[X,p]p = p[X,p]$ is actually a consequence of the other relations in θ. We include it order to satisfy the definition of type in Section 3.3, that if a variable (in this case $[X,p]$) appears in a type then there is a unique primitive word p with which it commutes.
2. Choose any $[X,p] \in \mathcal{X}_{\text{new}}$ and write $\sigma([X,p]) = p^e$. (Note that we have $e \geqslant 10$ in this case, since a meta rule would remove the variable if it has a solution shorter than 30δ). Define a morphism $\tau\colon M(B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \emptyset, \mu) \to M(B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu)$ by $\tau[X,p] = [X,p]p^5$. The morphism is well-defined since $[X,q]q = q[X,q]$ in $M(B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu)$ for all $[X,q] \in \mathcal{X}_{\text{new}}$.
3. Follow the corresponding substitution transition

$$(W, B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \emptyset, \mu) \xrightarrow{\varepsilon} (\tau(W), B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu').$$

The transition satisfies the forward property with the entire solution (α, σ') where $\sigma'[X,p] = p^{e-5}$. Apply all meta rules. (After that we may have $\theta = \emptyset$ again.)

(4) Rename $(\tau(W), \mu', \sigma')$ as (W, μ, σ).

endprocedure

Using the relations from θ we can move the $\mathcal{X}_{\text{new}}$ variables around over green positions. Thus, we can choose a word representation for $W \in M(B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu)$ as $W \in (B \cup \mathcal{Y} \cup \mathcal{X}_{\text{new}})^*$ such that every maximal green interval $[i,j]$ of positions in W is labeled by a word of the form

$$W[i,j] = [X_1, rs] \cdots [X_g, rs](rs) \cdots (rs)(rsr)\,[X_{g+1}, sr] \cdots [X_d, sr] \tag{21}$$

$$= [X_1, rs] \cdots [X_d, rs](rs) \cdots (rs)(rsr) \tag{22}$$

$$= (rsr)(sr) \cdots (sr)[X_1, sr] \cdots [X_d, sr]. \tag{23}$$

In the following we simply say that $W[i,j]$ is a *maximal green factor* when we actually refer to the label $W[i,j]$ of a maximal green interval $[i,j]$ of positions. We can choose $0 \leqslant d \leqslant 2$ because in a standard state all local equations are triangular, but this is not essential. Without restriction we have $r \neq 1$ and that $(rs) \cdots (rs)(rsr) = (rs)^e r$ satisfies $e \geqslant 1$. This is clear if $d = 0$. For $d \geqslant 1$ it is enough to substitute $[X_1, rs]$ by $[X_1, rs]rs$. For each $1 \leqslant i \leqslant d$ there is $e_i \in \mathbb{N}$ such that $\sigma[X_i, rs] = (rs)^{e_i}$ and therefore:

$$\sigma(W[i,j]) = (rs)^{e+e_1+\cdots+e_d}\, r = r(sr)^{e+e_1+\cdots+e_d}. \tag{24}$$

$$\sigma(\overline{W[i,j]}) = (\overline{r}\,\overline{s})^{e+e_1+\cdots+e_d}\, \overline{r} = \overline{r}(\overline{s}\,\overline{r})^{e+e_1+\cdots+e_d}. \tag{25}$$

At this point we will change θ to θ' (defined below) since the defining relations $[X, aq]a = a[X, qa]$ with $a \in B$ will be of no use anymore. The idea is to replace the factors rs, sr, and rsr by fresh letters denoted by $[rs]$, $[sr]$, and $[r, s, \lambda]$. The λ is used to encode the sum $e + e_1 + \cdots + e_d$. We will make the assumption $r \neq 1$, then $[rs]$, $[sr]$, and $[r, s, \lambda]$ are three different letters for $s \neq 1$, and there is no letter $[1, p_\lambda, \lambda]$, only $[p_\lambda, 1, \lambda]$. For the maximal green factor $W[i,j]$ we intend to define a word $W'[i,j]$ and a type θ' such that

$$W'[i,j] = [X_1, rs] \cdots [X_g, rs]\,[rs] \cdots [rs]\,[r, s, \lambda]\,[X_{g+1}, sr] \cdots [X_d, sr] \tag{26}$$

$$= [rs] \cdots [rs]\,[X_1, rs] \cdots [X_g, rs]\,[r, s, \lambda]\,[X_{g+1}, sr] \cdots [X_d, sr] \tag{27}$$

$$= [X_1, rs] \cdots [X_g, rs]\,[r, s, \lambda]\,[X_{g+1}, sr] \cdots [X_d, sr]\,[sr] \cdots [sr]. \tag{28}$$

More precisely, for each $u_\lambda p_\lambda^{e_\lambda}\, r_\lambda v_\lambda \in F_\lambda$ we associate a new letter $[r_\lambda, s_\lambda, \lambda]$ with $\mu'([r_\lambda, s_\lambda, \lambda]) = \mu(r_\lambda, s_\lambda r_\lambda)$, and $[q]$ for every typed variable $[X, q]$ with $\mu'[q] = \mu(q)$. Recall our notation that $u_\lambda p_\lambda^{e_\lambda}\, r_\lambda v_\lambda$ is a very long δ-periodic word, $|u_\lambda| = |v_\lambda| = 3\delta$, p is primitive, and $r \neq 1$. It is important that $[r, s, \lambda]$ is visible, whenever at least one the green positions is visible. This why in the different word representations (26)–(28) for the same $W'[i,j]$ the $[r, s, \lambda]$ always sits between the variables.

By introducing (if necessary) more fresh letters we close the set of fresh letters under involution and H-action. We let:

$$\overline{[r_\lambda, s_\lambda, \lambda]} = [\overline{r_\lambda}, \overline{s_\lambda}, \lambda] \text{ and } \overline{[q]} = [\overline{q}]. \tag{29}$$

$$g \cdot [r_\lambda, s_\lambda, \lambda] = [g \cdot r_\lambda, g \cdot s_\lambda, \lambda] \text{ and } g \cdot [q] = [g \cdot q] \text{ for } g \in H. \tag{30}$$

The set of these new letters is denoted by B_{new}. The number of new letters can be bounded by:

$$|B_{\text{new}}| \in \mathcal{O}(|H| \cdot (|W| + \delta n)).$$

Let $B' = B \cup B_{\text{new}}$. Next, we define the new type θ'. For each typed variable $[X, q]$ there is exactly one commutation rule: $[X, q]q = q[X, q]$. The other defining relations say that $[r_\lambda s_\lambda]$ and $[s_\lambda r_\lambda]$ are "conjugate" due to the letter $[r_\lambda, s_\lambda, \lambda]$. Making this formal we specify θ' by:

$$\theta' = \{([X,q][q],\ [q][X,q]) \mid [X,q] \in \mathcal{X}_{\mathrm{new}}\} \tag{31}$$

$$\cup \{[r,s,\lambda][sr],\ [rs]([r,s,\lambda] \mid [r,s,\lambda] \in B_{\mathrm{new}}\}\,. \tag{32}$$

Note that the defining relations in θ' are designed that (26)–(28) hold.

We can now define the rest of the δ-periodic compression procedure. It is the analogue to Jeż's "block compression" as described in [11]. During the process sets of positions for W and $\sigma(W)$ change, but our process makes clear that we can always transport the green color: no change involves an interval which has both colored and uncolored positions. We perform the following steps.

begin procedure (remove very long δ-periodic factors with a visible position)

(1) Define the element $W' \in M(B', \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta', \mu')$ by replacing maximal green factors in W just we as have done for $W[i,j]$ in Equation (21) in order to produce $W'[i,j]$ in Equation (26). Doing this everywhere defines W' in a word representation as $W' \in (B' \cup \mathcal{Y} \cup \mathcal{X}_{\mathrm{new}})^*$.

(2) Define a $B \cup \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}$-morphism

$$h_1 \colon M(B', \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta', \mu') \to M(B, \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta, \mu)$$

by $h[r,s,\lambda] = rsr$ and $h[q] = q$. We have $W = h_1(W')$ and we obtain a compression transition satisfying the forward property

$$E = (W, B, \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta, \mu) \xrightarrow{h_1} (W', B', \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta', \mu') = E'.$$

Note that $\|E'\| < \|E\|$ since $\|W'\| < \|W\|$ due to the fact that (by our assumption) at least one green interval exists with a visible position exists; and therefore some new letter $[r,s,\lambda]$ is visible in W' (which represents the word rsr of length at least 2). The new entire solution at E' is $(\alpha', \sigma') = (\alpha h_1, \sigma')$ where $\sigma'(X) = X$ for $X \in \mathcal{X}$ and $\sigma'[X,p] = [p]^e$ if $\sigma([X,p]) = p^e$. We apply the meta rules and then we rename E', W', α', σ' as E, W, α', σ' but we keep the notation for B', $\mathcal{X}$, $\mathcal{X}_{\mathrm{new}}$, θ', and μ' (although B', $\mathcal{X}_{\mathrm{new}}$, and θ' may become smaller by the meta rules and μ' changes).

(3) **while** there is letter $[p] \in B_{\mathrm{new}}$ **do**

(a) If $\mathcal{X}_{\mathrm{new}} \neq \emptyset$, then choose some $[X,p] \in \mathcal{X}_{\mathrm{new}}$. Use a substitution transition defined by $\tau[X,p] = [X,p][p]^2$ to make sure that $\sigma([X,p])$ is shorter than at the beginning of the loop and that we don't run out of letters $[p]$ as long as there are typed variables. The invariant is that as long as $\mathcal{X}_{\mathrm{new}} \neq \emptyset$ there is some letter $[p]$ visible.

(b) Use transitions of the form $[X,p] \mapsto [X,p][p]$ in order to keep the invariant that $\sigma([X,p]) = [p]^e$ where e is even. Moreover, due the meta rules we maintain $|\sigma([X,p])| \geqslant 30\delta$. At some point $|\sigma([X,p])|$ might be too short, then we remove $[X,p]$ from $\mathcal{X}_{\mathrm{new}}$. We also maintain the invariant that $|\sigma([X,p])| = |\sigma([X,q])| = |\sigma([\overline{X},\overline{q}])|$ for all p, q and $X \in \mathcal{X}$. Thus, if we remove one $[X,p]$, then all other typed variables using the symbol X are removed simultaneously and θ' becomes smaller, too.

(c) If there is a maximal green factor

$$W[i,j] = [X_1,p]\cdots[X_g,p][p]^e[r,s,\lambda]\,[X_{g+1},q]\cdots[X_d,q] \tag{33}$$

where $d \geqslant 0$ and e is odd, then define an endomorphism

$$h_\lambda \colon M(B', \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta', \mu') \to M(B', \mathcal{X} \cup \mathcal{X}_{\mathrm{new}}, \theta', \mu')$$

by $h_\lambda([r,s,\lambda]) = [p][r,s,\lambda]$. Thus, we can write

$$W'[i,j] = h_\lambda([X_1, p \cdots [X_m, p][p]^{e-1}[r,s,\lambda]\,[X_{m+1},q] \cdots [X_\ell, q]). \tag{34}$$

This defines a new equation W' and a B'-morphism σ' such that $h(W') = W$ and $\sigma h_\lambda(W') = \sigma'(W)$. Hence, there is a compression transition satisfying the forward property

$$E = (W, B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu) \xrightarrow{h_\lambda} (W', B', \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta', \mu') = E'.$$

As above, $\|E'\| < \|E\|$ since $\|W'\| < \|W\|$. The new entire solution at E' is $(\alpha', \sigma') = (\alpha h_2, \sigma')$. We apply the meta rules and then we rename E', W', α', σ' as E, W, α', σ' .

(d) Due to the previous steps: whenever we see a maximal green factor

$$[X_1, p] \cdots [X_g, p][p]^e[r,s,\lambda]\,[X_{g+1},q] \cdots [X_d, q],$$

then $\sigma[X_i, p] \in ([p][p])^*$ for $1 \leqslant i \leqslant d$ and e is even. Define an endomorphism

$$h_3 \colon M(B', \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta', \mu') \to M(B', \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta', \mu')$$

by $h_3([p]) = [p]^2$ for all $[p]$ which appear in B_{new}. Thus, we can write

$$W'[i,j] = h_3([X_1, p] \cdots [X_m, p][p]^{e/2}[r,s,\lambda]\,[X_{m+1},q] \cdots [X_\ell, q]). \tag{35}$$

This defines a new equation W' and a B'-morphism σ' such that $h(W') = W$ and $\sigma h(W') = \sigma'(W)$. Hence, there is a compression transition satisfying the forward property

$$E = (W, B, \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X} \cup \mathcal{X}_{\text{new}}, \theta', \mu') = E'.$$

As above, $\|E'\| < \|E\|$ since $\|W'\| < \|W\|$. The new entire solution at E' is $(\alpha', \sigma') = (\alpha h, \sigma')$. We apply the meta rules and then we rename E', W', α', σ' as E, W, α', σ' .

endwhile

endprocedure

It is clear that the procedure terminates in some standard state. Let us denote that state and its entire solution as:

$$E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s) \quad \text{and} \quad (\alpha_s, \sigma_s). \tag{36}$$

We began the routine at E_r. During the procedure we see symbols $[X,p]$, $[rs]$ and $[r,s,\lambda]$, and the length of the equation W grows as we pop out letters in the suffix and prefix of each variable. At the end all the new variables disappeared, either by the meta rules or when maximal green factors are compressed into a single letter $[r,s,\lambda]$. The only new letters in B_s are of the form $[r_\lambda, s_\lambda, \lambda]$ and there are not more than $|H| \cdot |W_r|$ of them.

The following proposition summarizes all the changes that happen in the procedure.

Proposition 10.3. *Let $E_r = (W_r, B_r, \mathcal{X}_r, \emptyset, \mu_r)$ be the state where we started δ-periodic-compression with an entire solution (α_r, σ_r). Let $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ denote the standard state with the entire solution (α_s, σ_s) where we finish δ-periodic compression and let $(W, B', \mathcal{X}', \theta, \mu)$ be any state which we have seen (with the full set of variables $\mathcal{Y}' = H \cdot \mathcal{X}'$) on the path from E_r to E_s during the procedure.*

Then we have the following.

(1) $B_s = B \cup \{[r_\lambda, s_\lambda, \lambda] \mid \lambda \in \Lambda\}$ for some $B \subseteq B_r$.
(2) $|B_s| \leqslant |H| \cdot (|W_r| + |W_s|)$.
(3) $|B'| \in |H| \cdot (|W_r| + \mathcal{O}(\delta n))$.
(4) $\|W_s\| \leqslant \|W_r\| + 14\delta n \leqslant \|W_r\| + 20\delta n$.

(5) $\|W\| \in \|W_r\| + \mathcal{O}(\delta n) \subseteq |W_r| + \mathcal{O}(\delta n)$.
(6) $\mathcal{X}_s \subseteq \mathcal{X}_r \subseteq \mathcal{X}$.
(7) $|\mathcal{X}'| \in \mathcal{O}(\delta n)$.
(8) *For each* $X \in \mathcal{X}_s$ *the word* $\sigma_s(X)$ *does not start or end with a very long* δ*-periodic word.*

Proof. We justify each item as follows.

(1) Meta rules may remove some (useless) letters from the initial alphabet B_r, so we have $B \subseteq B_r$, and the only new constants that survive at the end are of the form $\{[r_\lambda, s_\lambda, \lambda] \mid \lambda \in \Lambda\}$.

(2) B_s consists of letters in $H \cdot B_r$ and those from $H \cdot \{[r_\lambda, s_\lambda, \lambda] \mid \lambda \in \Lambda\}$. Since we applied alphabet reduction at the beginning, $|B_r| \leqslant H \cdot |W_r|$, and since the letters $[r_\lambda, s_\lambda, \lambda]$ cannot be eliminated by any compression during the procedure, their number is bounded by $H \cdot |W_s|$. The only other new constants added during the procedure are of the form $[p]$ but these are all eliminated by compression, so do not appear in B_s by the meta rule.

(3) Again we have $|B_r| \leqslant H \cdot |W_r|$. During the procedure we add letters $[rs]$ and $[r, s, \lambda]$. For each variable $X \in \mathcal{X}$ we add some $[rs]$ and $[r, s, \lambda]$, then we need to multiply by δ since we have all cyclic permutations of these and $|rs| \leqslant \delta$. Since the are at most n variables this gives $\mathcal{O}(\delta n)$ new constants, so applying the action we get the bound of $H \cdot \mathcal{O}(\delta n)$ new constants.

(4) We first pop out $\tau(X) = X[X,p]p'$ because we had $\sigma(X) = xq^dq' = xup^ep'$ with $|q^dq'| \geqslant 3\delta$. (If $|q^dq'| < 3\delta$ we do nothing.) After applying τ we have $\sigma(X) = xu$ and $\sigma([X,p]) = p^e$. If it is the case that $|p^ep'| < 7\delta$ then later we do not apply any compression to $[X,p]$ since it is not part of a very long factor, instead we simply pop it out. This contributes at most $14\delta n$ in the worst case that this happens in the suffix of every $X \in \mathcal{X}$.

If $|p^ep'| \geqslant 7\delta$ then together with u this gives a factor of length at least 10δ, so $[X,p]$ is part of a very long δ-periodic word so it is compressed down to a single letter. Thus 7δ is the most added at either end of any variable by the procedure. This gives $14\delta n$. We give the larger bound $20\delta n$ to simplify later calculations only.

(5) Since $\|W\| \leqslant |W| + 90\delta n \in |W| + \mathcal{O}(\delta n)$ at every state in $\mathcal{T}$, it is enough to show $\|W\| \in \|W_r\| + \mathcal{O}(\delta n)$. However this only requires the estimation in Section 10.1. By that estimation we content ourselves to define a function $s \colon \mathbb{N} \to \mathbb{N}$ with $s(0) \leqslant c\delta n$ and which satisfies for all t a bound

$$s(t+1) \leqslant q\, s(t) + c\delta n$$

for some $q < 1$ and $c \geqslant 1$. To see where the q comes from in our application, choose $s(0)$ to be number of letters $[p]$ at the state where they first appear. Each time we pass through a transition defined by $h([p]) = [p][p]$ we half the number of these letters; and this shows that we can define $q = 1/2$. Between these steps where we halve the number of $[p]$'s we create at most $c\delta n$ new ones with $c \in \mathcal{O}(1)$.

(6) During the procedure we add new variables $[X,p]$ but these are eliminated by the compression. Since we apply meta rules we may also remove variables $X \in \mathcal{X}_r$. Thus $\mathcal{X}_s \subseteq \mathcal{X}_r \subseteq \mathcal{X}$.

(7) This is justified above at Equation (20).

(8) Consider any $\sigma_s(X)$ with $X \in \mathcal{X}_s$. If for that X, the word $\sigma(X)$ at the beginning at the procedure "insert new variables" had a δ-periodic suffix of length more than 3δ, then, due to the splitting of variables, the length of the maximal δ-periodic suffix in $\sigma_s(X)$ is exactly 3δ. Hence, there is no very long δ-periodic suffix. In the other case the suffix of length 3δ in $\sigma(X)$

and $\sigma_s(X)$ coincide. Thus, in this case the length of the maximal δ-periodic suffix in $\sigma_s(X)$ is at most 3δ. Since $X \in \mathcal{X}$ implies $\overline{X} \in \mathcal{X}$ the same is true for the prefix of each variable.

The proposition is therefore shown. □

10.6. **The end of the δ-periodic compression.** Recall that we started a compression round at a standard state E_r with an equation $|W_r|$; and we end standard state E_s with an equation $|W_s|$. (Possibly $r = s$.) If, due the meta rules, E_s is final, we are done. Hence, we continue under the assumption that E_s is not a final state.

11. Pair compression

We enter the second phase of the compression round with "pair compression" directly after the end of δ-periodic compression. We enter the pair compression procedure at a standard state E_s which is not final and with an entire solution (α_s, σ_s). If E_r and (α_r, σ_r) denote the situation where we began the current compression round (where we began δ-periodic compression), then Proposition 10.3 tells us:

$$\|W_s\| \leqslant \|W_r\| + 20\delta n. \tag{37}$$

During the pair compression all states are standard states. No type is needed. Phrased differently, we have $\theta = \emptyset$, there are no typed variables and hence, all variables belong to $\mathcal{X}$. Thus, the number of positions labeled with twisted variables is at most n, as it is required for standard states.

Our goal in this section is to compress pairs $ab \leqslant W_s$ of constants into single letters without causing any conflict due to overlap with other pairs or variables that are connected via twisted equations. In particular, compressing a pair linked to a δ-periodic factor might cause problems, so we wish to avoid compressing those pairs. This leads us to define the following.

Definition 11.1. *Let $E = (W, B, \mathcal{X}, \emptyset, \mu)$ be a standard state with a solution σ. We say that (E, σ) satisfies the* shrinking pair condition *if there is no $X \in \mathcal{X}$ such the word $\sigma(X)$ starts with a very long δ-periodic word*[4].

Note this is the situation we find ourselves in at the conclusion of the δ-periodic compression procedure: $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ with its solution σ_s satisfies Definition 11.1. The shrinking pair condition is a necessary condition when proving Lemma 11.3 below. That technical lemma is one of the key steps.

11.1. **Positions revisited.** Consider any standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ together with a entire solution (α, σ). We introduce a precise notion of equivalence $\equiv$ between positions (and intervals) of $\sigma(W)$, which we introduce now. The idea is that whenever we modify a solution σ at a position i, then we must modify σ at all equivalent positions $j \equiv i$ in order to keep the property of being a solution. Moreover, $\equiv$ should be the finest equivalence relation with that property. For example, when we compress a factor ab where a, b are letters, then we want to compress only certain occurrences of $f(ab), f(\overline{b}\overline{a})$ (where $f \in H$) and not all of them.

As explained in Section 10.3 there is a canonical mapping from the set $[1, |\sigma(W)|]$ to the set of positions in W. By $\lambda(i)$ we denote the label of a position i in $\sigma(W)$. Recall also the notion of duality: if $[l, r]$ is an interval in $[1, |\sigma(W)|]$, then $\overline{[\ell, r]} = [\overline{r}, \overline{\ell}]$ denotes the dual interval, where $\overline{i} = |\sigma(W)| + 1 - i$ for all $1 \leqslant i \leqslant |\sigma(W)|$. According to the definition of standard states, the set of twisted variables which appear in W can be written as $\mathcal{Y} = (H \times \mathcal{X})$ and we have $\mathcal{X} \subseteq \mathcal{X}$. It is convenient

[4] Recall that a word w is very long δ-periodic if and only if $|w| \geqslant 10\delta$ and w is a prefix of some word $p^{|w|}$ where $|p| \leqslant \delta$, see Definition 6.3.

to fix a subinterval $I(X)$ of $[1,|\sigma(W)|]$ for each $X \in \mathcal{X}$ such that $\overline{I(X)} = I(\overline{X})$ as follows. Consider $\{X,\overline{X}\} \subseteq \mathcal{X}$, then without restriction we have $X = X_i$ for some unique $1 \leqslant i \leqslant |\mathcal{X}|/2$ (and hence $\overline{X} = \overline{X_i}$). Choose for $I(X)$ the left-most maximal interval $[\ell_X, r_X]$ in $\sigma(W)$ which is mapped to a position labeled by X_i. In particular, $r_X - \ell_X + 1 = |\sigma(X|$. We let $\overline{I(X)} = I(\overline{X})$. By the specific structure of W being an extended equation, we see that $\overline{I(X)}$ is the right-most maximal interval in $\sigma(W)$ which is mapped to a position labeled by $\overline{X_i}$. To have a notation we let $2m = |\sigma(W)|$ and

(38) $I(\mathcal{X}) = \bigcup \{I(X) \mid X \in \mathcal{X}\}$,

(39) $I(B) = \left\{i \in [1,2m] \;\middle|\; i \text{ or } \overline{i} \text{ is mapped to a position labeled by a constant}\right\}$.

Note that $I(\mathcal{X})$ and $I(B)$ are disjoint sets of positions: if $i \in I(\mathcal{X})$, then there is some X such that i is mapped to a position labeled by X and $\overline{i}$ is mapped to a position labeled by $\overline{X}$. The next idea is to identify positions in $\sigma(W)$ based on the fact that we can write W in the form $W = \#U\#\#\overline{V}\#$ such that $\sigma(W) = \sigma(\overline{W}) \iff \sigma(U) = \sigma(V)$. In pictures we intend to place the positions of $\sigma(U)$ and $\sigma(V)$ on top of each other. The intuition is clear: we have $[1,2m] = \left\{1,\dots,m,\overline{m},\dots,\overline{1}\right\}$. The positions of $\#\overline{V}\#$ cover $\overline{m},\dots,\overline{1}$. Hence, we can think that $\#V\# = \overline{\#\overline{V}\#}$ uses the same set of positions as $\#U\#$, namely the set $\{1,\dots,m\}$. Thus, every $i \in [1,2m]$ has always two interpretations: for $i \leqslant m$ as a position either in $\#U\#$ or in $\#V\#$, for $m < i$ the situation is dual. Let us make this intuition formal.

The mapping from $[1,2m]$ to the positions of W induces a relation

$$\rightsquigarrow \subseteq [1,2m] \times (I(\mathcal{X}) \cup I(B)).$$

We define $\rightsquigarrow$ inductively. If i is mapped to a position in W which labeled by a constant, then we have $i \in I(B)$ and we let $i \rightsquigarrow i$ and $\overline{i} \rightsquigarrow \overline{i}$. In the other case i is mapped to a position labeled by $Y = (f,X)$ for some $f \in H$ and $X \in \mathcal{X}$. Let ℓ the leftmost position in $[1,2m]$ which is mapped to the same position as i. Then we can write $i = \ell + k$ and we find $j = \ell_X + k$. In this case we let $i \rightsquigarrow j$ and $\overline{i} \rightsquigarrow \overline{j}$. Note that this implies $\lambda(i) = f(\lambda(j))$. The position $\overline{j}$ belongs to $I(\overline{X})$.

Up to duality, there are three cases:

1. $i \in I(B)$ and $\overline{i} \in I(B)$. Then there is only one j such that $i \rightsquigarrow j$.
2. $i \in I(B)$ and $\overline{i} \notin I(B)$. Then we have $i \rightsquigarrow i \in I(B)$ and $i \rightsquigarrow j \in I(\mathcal{X})$. Since $I(\mathcal{X}) \cap I(B) = \emptyset$, we have $i \neq j$.
3. $i \notin I(B)$ and $\overline{i} \notin I(B)$. Then $i \rightsquigarrow j$ and $\overline{i} \rightsquigarrow \overline{k}$ with $\left\{j,\overline{j},k,\overline{k}\right\} \subseteq I(\mathcal{X})$. Hence, there there are at most two $j,k \in I(\mathcal{X})$ such that $i \rightsquigarrow j$ and $i \leftsquigarrow k$.

Let us explain the meaning of this "$\rightsquigarrow$" relation by considering a local equation $u(f,X)w(g,\overline{Y})v = uZv$. In W this local equation corresponds to some factorization

$$W = u_1\#u(f,X)w(g,\overline{Y})v\#u_2\;\overline{u_2}\,\#\,\overline{v}\overline{Z}\overline{u}\,\#\,\overline{u_1}.$$

Let $\ell = |\sigma(u_1\#)| + 1$ and $r = |\sigma(XwYv| - 1$. Then the interval $\sigma(W)[\ell,r]$ is labeled by $u\sigma(f,X)w(g,\overline{Y})v$; and, since σ is a solution, we have $u\sigma(f,X)w(g,\overline{Y})v = u\sigma(Z)v$. For each $i \in [\ell,r]$ we have $i \rightsquigarrow j$ where either $i = j$ or $j \in I(X) \cup I(\overline{Y})$. If $j \in I(X) \cup I(\overline{Y})$, then $\overline{i} \rightsquigarrow \overline{k} \in I(\overline{Z})$ and therefore $i \rightsquigarrow k \in I(Z)$, too. The positions of u, w, v are visible in W, but with respect to $\overline{W}$ this true only for the positions of u and v. For the positions of u and v the relation $\rightsquigarrow$ is the identity. The relations are depicted in Figure 5: the relation $\rightsquigarrow$ is given by the positions in the middle row to the top and bottom row. Let $j \leftsquigarrow i$ denote $i \rightsquigarrow j$. If $j \leftsquigarrow i \rightsquigarrow k$, then we write $j \sim k$.

Consider any $i \in [1,2m]$ and $j,k \in I(\mathcal{X}) \cup I(B)$ such that $j \leftsquigarrow i \rightsquigarrow k$. (The interesting case is $j \neq k$.) Hence, we have $j \sim k$ and in the pictures we put the positions j and k on top of each other. Figure 5 gives an example, where $32 \sim 10, \dots, 35 \sim 13$, $98 \sim 14, 99 \sim 15$, and $\overline{45} \sim 16, \dots, \overline{42} \sim 19$.

		$I(X)$						$I(\overline{Y})$					
a	a	32	33	34	35	a	b	$\overline{45}$	$\overline{44}$	$\overline{43}$	$\overline{42}$	b	b
92	93	94	95	96	97	98	99	100	101	102	103	104	105
a	a	10	11	12	13	14	15	16	17	18	19	b	b
		$I(Z)$											

FIGURE 5. A local equation $bb(f,X)ab(g,\overline{Y})aa = bbZaa$. The left side occupies positions $92, \dots, 105$ and dual of the right side (i.e. $\overline{b}\,\overline{b}\,\overline{Z}\,\overline{a}\,\overline{a}$) positions $\overline{105}, \dots, \overline{92}$. Moreover, $\sigma(W)[92,93] = aa$, $\sigma(W)[98,99] = ab$, and $\sigma(W)[104,105] = bb$.

Since $i \rightsquigarrow j \iff \overline{i} \rightsquigarrow \overline{j}$ we have

$$j \sim k \iff \overline{j} \sim \overline{k}. \tag{40}$$

Moreover, $j \sim k$ implies that there are $f, g \in H$ such that $f(\lambda(j)) = \lambda(i) = g(\lambda(k))$. Thus, $j \sim k$ ensures $\lambda(j) \in H \cdot \lambda(k)$.

We have $I(\mathcal{X}) \cup I(B) \subseteq [1, 2m]$ and $\sim \subseteq [1,2m] \times [1,2m]$ is a symmetric relation. It is also reflexive on $I(\mathcal{X}) \cup I(B) \times I(\mathcal{X}) \cup I(B)$.

By $\approx$ we denote the reflexive and transitive closure of $\sim$. However, the relation $\approx$ is too fine, in general. Since $i \sim j$ implies $\lambda(i) \in H \cdot \lambda(j)$, we cannot expect that $i \approx \overline{i}$, because $\lambda(\overline{i}) = \overline{\lambda(i)}$ is typically not in $H \cdot \lambda(j)$. Clearly, if we intend to change the label at position i from, say, a to c, then we must change the label at position $\overline{i}$ from $\overline{a}$ to $\overline{c}$.

In the following, we write $i \leftrightarrow j$ if $j = \overline{i}$ and we define $\equiv$ to be the equivalence relation over $[1, 2m]$ which is generated by $\sim \cup \leftrightarrow$. We have $\approx \subseteq \equiv$, but we have just seen that these relations are different, in general. Since $i \sim j \iff \overline{i} \sim \overline{j}$ by Equivalence (40), we have

$$i \leftrightarrow \overline{i} \sim j \iff i \sim \overline{j} \leftrightarrow j.$$

Hence,

$$i \equiv j \iff \text{either } i \approx j \text{ or } i \approx \overline{j}. \tag{41}$$

We extend the notation above to intervals. Let i, j be positions in $\sigma(W)$ such that $i \rightsquigarrow j$, and let $p \in \mathbb{N}$. Assume (by symmetry) that we have $i \rightsquigarrow j \in I(X)$ due to mapping the position i of $\sigma(W)$ to a position q in W which is labeled by a twisted variable (f, X). If the position $i + p$ is also mapped to the same position q, then all positions in the interval $[i, i+p]$ are mapped to q. We then write

$$[i, i+p] \rightsquigarrow [j, j+p].$$

As above we now define the relation $\sim$ on intervals. Again, we let $\approx$ be the generated equivalence relation, now on intervals. Finally, we relate an interval $[l, r]$ with $1 \leqslant l \leqslant r \leqslant 2m$ to the interval $[\overline{r}, \overline{l}]$ via

$$[l, r] \leftrightarrow \overline{[l, r]} = [\overline{r}, \overline{l}].$$

Thus, we also extend $\equiv$ to the equivalence relation on intervals which is generated by $\sim$ and $\leftrightarrow$. Having this, the general form of (41) becomes

$$[i, i+p] \equiv [j, j+p] \iff [i, i+p] \approx [j, j+p] \vee [i, i+p] \approx [\overline{j+p}, \overline{j}]. \tag{42}$$

In the following we say that two positions (or intervals) are *equivalent* if they are related by $\equiv$.

Lemma 11.2. *Let $t \geqslant 1$ and σ be a solution at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ such that $|\sigma(X)| \geqslant 2t$ for all $X \in \mathcal{X}$. For each $X \in \mathcal{X}$ write $\sigma(X) = uxv$ with $|u| = |v| = t$. Let $E' = (W', B, \mathcal{X}', \emptyset, \mu')$ and σ' denote state and solution which we obtain by following a substitution transition defined by $\tau(X) = uXv$ for all X. Let $I'(\mathcal{X}')$ be the set of positions according to Equation (38) and $\approx'$ the equivalence relation on intervals with respect to σ'. Let i and j positions in $I'(\mathcal{X}')$ such that $i \approx' j$, then we have $[i-t, i+t] \approx [j-t, j+t]$ with respect to E and σ.*

Proof. Let $X \in \mathcal{X}'$. Then $X \in \mathcal{X}$ and if $I(X) = [\ell_X, r_X]$, then the corresponding interval $I'(X)$ with respect to σ' is $I'(X) = [\ell_X + t, r_X - t]$. Thus, if $i \approx' j$ with $i \in I'(X)$ and $j \in I'(Y)$, then we obtain a "domino tower" as depicted in Figure 6. The intervals with respect to $I'(\mathcal{X}')$ are the white blocks and i and j are arranged such that i sits along a vertical column above j. We obtain the corresponding tower with respect to $I(\mathcal{X})$ by adding the grey borders of length p at each side in order to switch from $I'(Z)$ to $I(Z)$. Thus, $i \approx j$ with respect to W' implies $[i-t, i+t] \approx [j-t, j+t]$. □

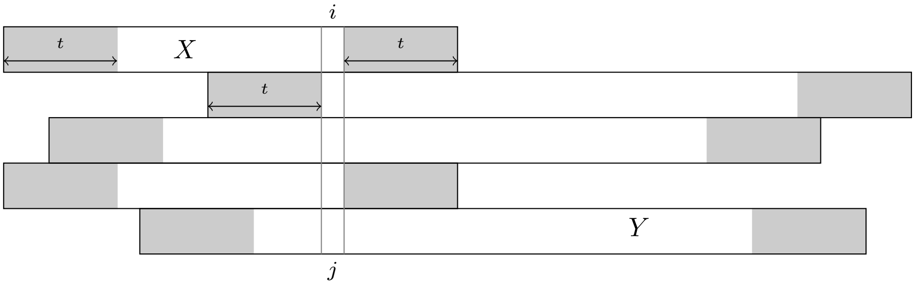

Figure 6. Example illustrating Lemma 11.2.

The *distance* $d(i,j)$ between positions of $\sigma(W)$ is denoted as usual: $d(i,j) = |j - i|$. We continue with the same the notation as in Lemma 11.2. We apply the lemma with $t = 10\,|H|\,\varepsilon$. Recall that we fixed the parameters such that $\varepsilon = 30n$ and $\delta = |H|\,\varepsilon$. Hence, $t = 10\delta$.

Lemma 11.3. *Let $t = 10\delta$ and suppose that $E = (W, B, \mathcal{X}, \emptyset, \mu)$ satisfies the shrinking pair condition (Definition 11.1) with respect to σ. Let $Z \in \mathcal{X}'$ be a variable with $I'(Z) = [l, r]$ (and hence, $I(Z) = [l - 10\delta, r + 10\delta]$); and let $l \leqslant i < j \leqslant r$ such that $i \approx' j$ with respect to σ'. Then we have $d(i,j) > \varepsilon$.*

Proof. According to Lemma 11.2 we have $[i-t, i+t] \approx [j-t, j+t]$ with respect to σ. Next, we choose $k \in \mathbb{N}$ as large as possible so that $[i-k, i+t]$ is a subinterval of $I(Z)$ and $[i-k, i+t] \approx [j-k, j+t]$ with respect to σ. By induction on the number of steps using the relation $\sim$, this implies that there is some interval $[\ell, \ell + t + k] \subseteq I(X')$ such that both, first

$$[j-k, j+t] \approx [i-k, i+t] \approx [\ell, \ell + t + k]$$

and second, ℓ is the first position in $I(X')$, see Figure 7. Assume, by contradiction, $d(i,j) \leqslant \varepsilon$. Then, $\sigma(W)[i-k, i+t]$ and $\sigma(W)[j-k, j+t]$ are twisted conjugate with a positive offset $d(i,j)$ which is at most ε. This implies that $\sigma(W)[i-k, i+t]$ is a very long δ-periodic word by Corollary 6.2. Therefore, the prefix $\sigma(W)[\ell, \ell + t + k]$

of $\sigma(X')$ is a very long δ-periodic word. This contradicts the hypothesis that E satisfies the shrinking pair condition. □

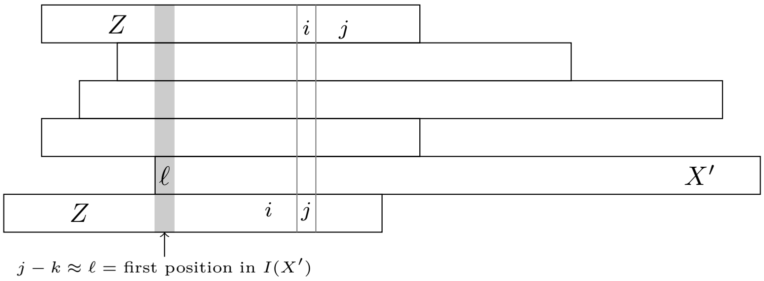

FIGURE 7. Illustration for the proof of Lemma 11.3.

11.2. **Red positions.** We use the notation of Section 11.1. Let $[l, r]$ be a maximal interval in $\sigma(W)$ which is mapped to a position in W which is labeled by some (f, X) where $f \in H$ and $X \in \mathcal{X}$. Thus, the factor $\sigma(W)[l, r]$ is equal to the word $\sigma(X)$. The positions l and r at the borders of $[l, r]$ play a special role because if there is a factor $ab = \sigma(W)[l-1, l]$, then we cannot compress ab because ab is "crossing": compressing ab might be "dangerous". In order to signal "danger" we color the first and the last position in each interval $I(X)$ *red*. Moreover, whenever $i \equiv j$ holds in $\sigma(W)$, then color j *red*, too. For example, if we have a situation as depicted in Figure 8, then the red color at the last position of $I(X)$ and the red color at the first position of $I(Z)$ yields two red columns. Note that the first red position in $I(X)$ and the last red position in in $I(\overline{X})$ are equivalent: these are dual positions. For convenience, we also color all positions in $\sigma(W)$ *red* which are labeled by the marker symbol #. If the label of i is #, then $i \equiv j \iff j \in \{i, \bar{i}\}$.

Since $i \equiv \bar{i}$ for all positions, it follows that there are at most n pairwise different equivalence classes of red positions. This counting will be used later.

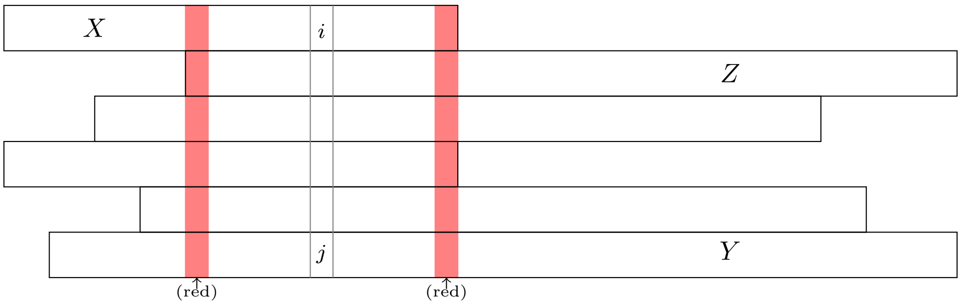

FIGURE 8. Red positions induced by the red borders in $I(X)$ and $I(Z)$.

Consider an interval of length two $I = [i, i+1]$ without red position. The idea is to compress I into a single position. The problem is overlapping: we might have $[i-1, i] \equiv [i, i+1]$ or $[i, i+1] \equiv [i+1, i+2]$. Note that $[i-1, i] \equiv [i, i+1]$ implies $i-1 \equiv i$ or $i-1 \equiv i+1$. Similarly, $[i, i+1] \equiv [i+1, i+2]$ implies $i \equiv i+1$ or $i \equiv i+2$. Therefore, we start with intervals of length 4 where all four positions are inequivalent: This enables us to compress the middle interval of length 2. We shall use the following lemma.

Lemma 11.4. *Let $[i-1, i, i+1, i+2]$ be an interval of length 4 without any red position and where the positions are pairwise inequivalent. Consider $[i, i+1] \equiv [j, j+1] \equiv [k, k+1]$. Then there are two cases:*

(1) $[j, j+1] \cap [k, k+1] = \emptyset$,
(2) $k = j$ and $[j, j+1] \not\approx [\overline{k+1}, \overline{k}]$.

Proof. Notice that each of the intervals $[i-1, i, i+1, i+2]$, $[j-1, j, j+1, j+2]$, $[k-1, k, k+1, k+2]$ is without red positions. We may assume that $[j, j+1] \cap [k, k+1] \neq \emptyset$ because otherwise we are done.

First, let $j = k$. By contradiction assume $[j, j+1] \approx [\overline{k+1}, \overline{k}]$. Then $j+1 \approx \overline{k} \leftrightarrow k$ which implies $k = j \equiv j+1$. Since $[i, i+1] \equiv [j, j+1]$ we obtain $i \equiv i+1$. This was excluded.

In the second case we have $j \neq k$. Let us show that $j \neq k$ and $[j, j+1] \cap [k, k+1] \neq \emptyset$ leads again to a contradiction. Since $j \neq k$ we cannot have $[j, j+1] = [k, k+1]$. Hence $j+1 = k$ or $k+1 = j$. By symmetry in j and k, we may assume $j+1 = k$.

We cannot have $[j, j+1] \approx [k, k+1]$ because then $j \equiv k$, but $k = j+1$, and hence, $j \equiv j+1$. This is impossible. Thus, $[j, j+1] \approx [\overline{k+1}, \overline{k}]$ and $j \equiv k+1 = j+2$. We remember $j \equiv j+2$. If $[i, i+1] \approx [j, j+1]$, then (as no position is red) $[i, i+1, i+2] \approx [j, j+1, j+2]$ implies $i \approx i+2$. This is impossible. Hence, the last option is $[\overline{j}-1, \overline{j}] \approx [i, i+1]$ and $[\overline{j}-2, \overline{j}-1, \overline{j}] \approx [i-1, i, i+1]$. However, $j \equiv j+2$ implies $\overline{j} \equiv \overline{j}-2$. We have again a contradiction as $i-1 \not\equiv i+1$. □

Example 11.5 indicates why the assertion in Lemma 11.4 only holds in the middle interval $[i, i+1]$ of $[i-1, i, i+1, i+2]$, in general.

Example 11.5. *We don't exclude that H acts with involution. Thus, there might be an $a \in B$ and $f \in H$ such that $f(a) = \overline{a}$. Consider the equation $\overline{X} = (f, X)$ with the solution $\sigma(X) = dcba\overline{b}\overline{c}\overline{d}$ and where $f(x) = x$ for $x = b, c, d$. Then we have*

$$\sigma(\overline{X}) = dcb\overline{a}\overline{b}\overline{c}\overline{d} = f(\sigma(X)).$$

The positions of $\sigma(X)$ can be identified with $\{1, \ldots, 7\}$ with $i \equiv 8-i$ for all positions $1 \leqslant i \leqslant 7$. Since positions 3 and 5 are equivalent, the interval $[2, 5]$ contains equivalent positions. The four positions in the interval $[1, 4]$ are pairwise inequivalent. However, $[3, 4]$ intersects with $[4, 5] = [4, \overline{3}]$. Thus, later on we cannot compress the interval $[3, 4]$ corresponding to the pair ba. On the other hand, there is no obstacle to compress the interval $[2, 3]$ which is labeled by cb.

Lemma 11.6. *Let σ be a solution at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ and $I = [p, p+9]$ be an interval of length 10 in $\sigma(W)$ without any red position such that $i \approx j$ implies $i = j$ for all $i, j \in I$. Then I contains a subinterval Q of length 4 where all positions are pairwise inequivalent.*

Proof. For simplicity of notation let $I = [1, 10]$. If all positions in $[4, 5, 6, 7]$ are pairwise inequivalent, we are done. In particular, there are $1 \leqslant i < j \leqslant 10$ such that $i \equiv j$. Since, $i \not\approx j$ this implies $i \approx \overline{j}$ by (41). This in turn means that we cannot have $i \equiv j \equiv k$ where $1 \leqslant i < j < k \leqslant 10$ because this would lead to $i \approx k$ via $i \approx \overline{j} \approx k$. We say that j is the partner of i if $i \neq j$ but $i \equiv j$. We conclude that every $i \in I$ has at most one partner.

We know $4+i \equiv 4+j$ for some $0 \leqslant i < j \leqslant 3$. Hence $4+i \approx \overline{4+j}$. Let $1 \leqslant k \leqslant 3$. Since I is without red positions, this implies

$$4+i-k \approx \overline{4+j}-k = \overline{4+j+k} \leftrightarrow 4+j+k \equiv 4+i-k.$$

This means that every position $q \in Q = \{4+i-3, 4+i-2, 4+i-1, 4+i\}$ has one partner in $P = \{4+j+3, 4+j+2, 4+j+1, 4+j\}$. Since $Q \cap P = \emptyset$ and $Q \cup P \subseteq I$, we are done: in Q all four positions are pairwise inequivalent. □

Definition 11.7. *We say that an interval $I = [i, i+1]$ in $\sigma(W)$ is* good *if the following conditions hold.*

- *Neither i nor $i+1$ is red.*
- *The positions i and $i+1$ are visible,*
- *Whenever $[i, i+1] \equiv J \equiv K$, then either $J = K$ or $J \cap K = \emptyset$.*

Remark 11.8. *The definition of a good interval $I = [i, i+1]$ excludes $i \approx i+1$. Indeed, since neither i nor $i+1$ is red, $i \approx i+1$ implies $[i-1, i, i+1] \approx [i, i+1, i+2]$, hence an overlap $[i, i+1] \approx [i+1, i+2]$. However, $I \approx [\overline{i+1}, \overline{i}]$ is allowed. Hence, it may happen that $i \equiv i+1$. If such an I is labeled with ab, then there is some $f \in H$ with $f(ab) = \overline{b}\overline{a}$. Hence, $f(a) = \overline{b}$ and $f(b) = \overline{a}$. We deduce that we obtain a consistent labeling for*

$$I = [i, i+1] \approx [\overline{i+1}, \overline{i}] \leftrightarrow [i, i+1].$$

Thus, we may compress $[i, i+1]$ into a single position with label c and therefore, due to $I \approx [\overline{i+1}, \overline{i}]$, we must also compress $[\overline{i+1}, \overline{i}]$ into a single position with label $f(c)$; and due to $[\overline{i+1}, \overline{i}] \leftrightarrow [i, i+1]$, we have to compress $[\overline{i+1}, \overline{i}]$ into a single position with label $\overline{c}$. But this is fine, we have $h(c) = ab$, $h(\overline{c}) = \overline{b}\,\overline{a}$, and $f(c) = \overline{c}$.

11.3. **The procedure.** Recall that we have fixed $\delta = |H|\,\varepsilon$ and $\varepsilon = 30n$. Thus, $\delta \in \mathcal{O}(|H|\,n)$ and $\varepsilon \in \mathcal{O}(n)$. We start at a standard state $E = (W, B, \mathcal{X}, \emptyset, \mu)$ together with an entire solution (α, σ) where none of the meta rules apply. In particular, $|\sigma(X)| \geqslant 30\delta$ and $\mathcal{X} \subseteq \mathcal{X}$ with $\mathcal{Y} = H \times \mathcal{X}$. Hence, by definition:

$$\sum_{Y \in \mathcal{Y}} |W|_Y \leqslant \sum_{Y \in \mathcal{X}} |W_{\text{init}}|_Y \leqslant n.$$

All local equations have the form $u(f, X)w(g, Y)v = uZv$. (As before dummy variables are allowed.) We define the equivalence relations $\approx$ and $\equiv$ over the set of positions of $\sigma(W)$ as defined in Section 11.1. Let E be a standard state with equation W and entire solution (α, σ). Once, we found a good interval I in $\sigma(W)$, we may call the following procedure for that interval.

begin procedure (compress a good interval I)

(1) Let $a, b \in B$ and let ab be the label of the good interval $I = [i, i+1]$. Choose a fresh letter c with stabilizer $H_c = H_a \cap H_b$; and define a B-morphism from $B' = B \cup \{f(c), f(\overline{c}) \mid f \in H\}$ to B^2 by $h(c) = ab$. Whenever $[i, i+1] \approx [j, j+1]$, then the label of $[j, j+1]$ is $f(ab)$ for some $f \in H$. Replace each of the intervals $[j, j+1]$ (resp. $[\overline{j+1}, \overline{j}]$) by a single new position and label this position with $f(c)$ (resp. $f(\overline{c})$). (There is no conflict in this relabeling, see Remark 11.8.) Since there is no red position in $[j, j+1]$ and $[\overline{j-1}, \overline{j}]$, none of the intervals $[j, j+1]$ or $[\overline{j-1}, \overline{j}]$ is "crossing". So, this gives a new but shorter equation W'. We have $h(W') = W$ and new solution σ' such that $h\sigma'(W') = \sigma(W)$.

(2) Follow the corresponding compression transition

$$E = (W, B, \mathcal{X}, \emptyset, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \emptyset, \mu') = E'.$$

We have a new state E' with an entire solution $(\alpha', \sigma') = (\alpha h, \sigma')$. There is also new numbering for the positions, but the red positions can still be identified.

```
endprocedure
```

We are now ready to define the procedure “pair compression” which uses “compress a good interval” as a subroutine.

```
begin procedure (pair compression)
  (1) For every $X \in \mathcal{X}$ write $\sigma(X) = uxv$ with $|u| = |v| = 10\delta$. Follow a
      substitution transition x
          $E = (W, B, \mathcal{X}, \emptyset, \mu) \xrightarrow{\varepsilon} (W', B, \mathcal{X}, \emptyset, \mu') = E'$
      defined by the substitution $\tau(X) = uXv$ and $W' = \tau(W)$. This transition
      satisfies the forward property with the new entire solution $(\alpha, \sigma')$ where
      $\sigma'(X) = x$ for all $X \in \mathcal{X}$. Recall that $\sigma(W) = \sigma'(W')$.
        After the preceding step, define the intervals $I'(X)$ with respect to $\sigma'$ as
      done in Section 11.1. Use the red color for the first and the last position in
      each $I'(X)$. Color in red all equivalent positions in $\sigma'(W')$ of red positions
      with respect to $\equiv'$, too. See Section 11.2.
  (2) Rename $E', W', \mu', \sigma', \approx', \equiv'$ as $E, W, \mu, \sigma, \approx, \equiv$.
  (3) Define the alphabet $B_{\text{old}} = B$. During the following loop we keep the
      invariant $B_{\text{old}} \subseteq B$.
  (4) while $\sigma(W)$ contains a good interval $I = [i, i+1]$ with a label in $B_{\text{old}}^2$
      do
        (a) Choose any good interval $I$ in $\sigma(W)$.
        (b) Run the procedure “compress $I$”.
        (c) Rename $E', B', W', \mu', \alpha', \sigma', \approx', \equiv'$ as $E, B, W, \mu, \alpha, \sigma, \approx, \equiv$ and trans-
            fer the induced coloring of red positions.
      endwhile
  (5) Perform an alphabet reduction at the standard state $E$.
  (6) Rename $E, B, \mathcal{X}, W, \mu, \alpha, \sigma$ as $E_{r'}, B_{r'}, \mathcal{X}_{r'}, W_{r'}, \mu_{r'}, \alpha_{r'}, \sigma_{r'}$.
endprocedure
```

Remark 11.9. *The procedure “pair compression” may not actually succeed in compressing any pair. Its first step always “pops out” letters to make the equation longer (by 20δ). After that if no pair is compressed, the procedure leaves the equation longer than before it was called. This is intentional: if the equation becomes long enough, then one of δ-periodic- or pair compression is guaranteed to reduce the equation size by a positive fraction.*

11.4. The end of pair compression ends the compression round.

We began the compression round at a standard state $E_r = (W_r, B_r, \mathcal{X}_r, \emptyset, \mu_r)$ with an entire solution (α_r, σ_r). We ended the δ-periodic compression either by entering a final state or, in the other case, at a standard state $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ with an entire solution (α_s, σ_s) such that

$$\|E_s, \alpha_s, \sigma_s\| \leqslant \|E_r, \alpha_r, \sigma_r\| \text{ and } \|W_s\| \leqslant \|W_r\| + 20\delta n. \tag{43}$$

We started the pair compression at the standard state $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ with the entire solution (α_s, σ_s). Compression took place only for good intervals which where labeled by words ab with $a, b \in B_{\text{old}} = B_s$. Each compression reduced the length of the equation because a good interval consists of two visible positions. Thus, at most $|W_s|$ compressions were possible; and this shows that we did not introduce more than $|H| \cdot |W_s| \in |H| \cdot |W_r| + \mathcal{O}(\delta n)$ fresh letters. Thus, every alphabet B of constants we met during the entire round satisfied

$$|B| \in |H| \cdot |W_r| + \mathcal{O}(\delta n). \tag{44}$$

Now, let $E_{r'} = (W_{r'}, B_{r'}, \mathcal{X}_{r'}, \emptyset, \mu_{r'})$ denote the standard state with the entire solution $(\alpha_{r'}, \sigma_{r'})$ where we end the procedure "pair compression". In the very first step of the procedure we followed a substitution transition. This is enough to infer

$$\|E_{r'}, \alpha_{r'}, \sigma_{r'}\| < \|E_s, \alpha_s, \sigma_s\| \leqslant \|E_r, \alpha_r, \sigma_r\| . \tag{45}$$

Proposition 11.10. *Let $E_r = (W_r, B_r, \mathcal{X}_r, \emptyset, \mu_r)$ be a standard state with an entire solution (α_r, σ_r) at the start of a compression round and $E_{r'} = (W_{r'}, B_{r'}, \mathcal{X}_{r'}, \emptyset, \mu_{r'})$ be the standard state where we end the round with the entire solution $(\alpha_{r'}, \sigma_{r'})$. Then we have*

$$\|W_{r'}\| \leqslant \frac{29\,\|W_r\|}{30} + \mathcal{O}(\delta n).$$

Moreover, if W is any equation which we see on the path from E_s to $E_{r'}$, then we have $\|W\| \leqslant \|W_r\| + \mathcal{O}(\delta n)$.

Proof. Each compression round has two phases. The δ-periodic compression stops at a standard state $E_s = (W_s, B_s, \mathcal{X}_s, \emptyset, \mu_s)$ with an entire solution (α_s, σ_s). By Proposition 10.3 $\|W_s\| \leqslant \|W_r\| + 20\delta n$ and all intermediate equations W satisfy $\|W\| \leqslant \|W_r\| + \mathcal{O}(\delta n)$.

Now, let W be any equation being on the path from E_s to $E_{r'}$. The additional length for W is due to the first step in pair compression when we substitute variable X by $u_X X v_X$ with $|u_X| = |v_X| = 10\delta n$. This shows $\|W\| \leqslant \|W_r\| + 40\delta n$. Moreover, $\|W_{r'}\| \in |W_r| + \mathcal{O}(\delta n)$ and $|W_s| \leqslant \|W_s\|$.

Thus, by Lemma 10.1 it suffices to prove

$$|W_{r'}| \leqslant \frac{29\,|W_s|}{30} + \mathcal{O}(\delta n). \tag{46}$$

Let us have a closer look at a local equation $u(f,X)w(g,Y)v = uZv$ in W_s. (We allow dummy variables). In particular, we can think that W_s begins with a prefix $U\#u_1Z_1v_1\#$ and ends with a suffix $\#\overline{v_1}\,\overline{Z_1}\,\overline{v_1}\#\overline{U}$. Having this, the word W_s is covered by factors $\#u(f,X)w(g,Y)v\#$ and $\#\overline{v}\overline{Z}\overline{u}\#$. In the first steps of pair compression we follow substitution transitions and a factor $\#u(f,X)w(g,Y)v\#$ becomes $\#uf(u_X)(f,X)f(v_X)wg(u_Y)(g,Y)g(v_Y)v\#$ and $\#\overline{v}\overline{Z}\overline{u}\#$ becomes $\#\overline{v}\,\overline{v_Z}\,\overline{Z}\,\overline{u_Z}\,\overline{u}\#$.

Pair compression compresses all factors $uf(u_X)$, $g(v_Y)v$, $\overline{v}\,\overline{v_Z}$ and $\overline{u_Z}\,\overline{u}$ into single letters. This bounds the total increase by the first substitution transitions by $2n$.

We don't have such a simple bound for the factors $f(v_X)wg(u_Y)$ because the corresponding positions in $\sigma(W_s)$ interact with the positions in $I(Z)$. Let $[\ell, r]$ be the interval in $[1, |\sigma(W_s)| - 1]$ corresponding to $f(v_X)wg(u_Y)$. Let us cut the interval in $[\ell, r]$ into a disjoint union of intervals, each of them having exact length $\varepsilon = 30n$. If a position belongs to any of these intervals of length $\varepsilon = 30n$, then we mark the position. Thus, at least $|w| - \varepsilon$ positions in the interval $[\ell, r]$ belonging to $f(v_X)wg(u_Y)$ are marked by these intervals. (We have no better bound since X and Y might be dummy variables.) Removing if necessary at most n of these intervals we may assume that their total number is ℓn with $\ell \geqslant 0$. The crucial observation is that we have

$$|\,|W_s| - \ell\varepsilon n| \in \mathcal{O}(\delta n). \tag{47}$$

Each interval of length ε is split in $3n$ intervals of length 10. By Lemma 11.3 an interval of length ε can have at most $2n$ red positions. Thus, in each interval of length ε there are at least n intervals of length 10 without any red position. By Lemma 11.6 each such interval $[i, i+9]$ contains an interval of length 4 where all positions are inequivalent. By Lemma 11.4 we can compress at least one interval in that interval of length 10. (Note that Lemma 11.4 provides us with a compression inside $[i+1, i+8]$. This means, the compression is guaranteed even if we compressed before an interval $[i-1,i]$ or $[i+9,i+10]$.) This means that the length of $\varepsilon = 30n$ is reduced to at most $\varepsilon = 29n$ during compression. Hence,

$$|W_{r'}| \in \frac{29}{30}\ell\varepsilon n + \mathcal{O}(\delta n). \tag{48}$$

Due to (47) we conclude $|W_{r'}| \in |W_s| + \mathcal{O}(\delta n)$. This shows (46) and hence, the assertion of the proposition. □

Remark 11.11. *Proposition 11.10 tells us that there is a constant $\kappa_1 \in \mathbb{N}$ such that $\|W_{r'}\| \leqslant \frac{29\|W_r\|}{30} + \kappa_1\delta n$. We content ourselves with a generous bound by letting $\kappa_1 = 97$. This bound suffices and it is an overestimation, as it can seen by the preceding proof and by reversing the $\mathcal{O}$-notation into concrete constants. The value $\kappa_1 = 97$ was chosen such that the later constant κ in Corollary 12.1 is divisible by 100. Thus, we can conclude $W_{r'}$ and for every equation W we see on the path from E_r to $E_{r'}$ the following upper bounds.*

(1) If $\|W_r\| > 30\kappa_1\delta n$, then $\|W_{r'}\| < \|W_r\|$ and $\|W\| < \|W_r\| + 40\delta n$.
(2) If $\|W_r\| \leqslant 30\kappa_1\delta n$, then $\|W\| \leqslant \|W_r\| + 40\delta n \leqslant 2960\delta n$.

These estimations are used in next section.

12. Putting it all together: the overall compression method

Now we explain what we do if we start the first compression round at the initial state E_{init} with a given initial entire solution $(\mathrm{id}_{A^*}, \sigma_{\text{init}})$. We begin a first compression round r with $r = 0$ and $E_0 = E_{\text{init}}$ with a given initial entire solution $(\alpha_0, \sigma_0) = (\mathrm{id}_{A^*}, \sigma_{\text{init}})$. We end the round after one phase each of δ-periodic compression and pair compression with a standard state E_1 and an entire solution (α_1, σ_1) such that $\|E_1, \alpha_1, \sigma_1\| < \|E_0, \alpha_0, \sigma_0\|$. We repeat this process by starting the next round $r+1$ with E_r and (α_r, σ_r) and ending the round in $E_{r'}$ and $(\alpha_{r'}, \sigma_{r'})$. For simplicity of notation we write $r + 1 = r'$. Thus,

$$E_{r'} = E_{r+1} = (W_{r+1}, B_{r+1}, \mathcal{X}_{r+1}, \emptyset, \mu_{r+1}) \text{ and } (\alpha_{r'}, \sigma_{r'}) = (\alpha_{r+1}, \sigma_{r+1}).$$

We conclude

$$\|E_{r+1}, \alpha_{r+1}, \sigma_{r+1}\| < \|E_r, \alpha_r, \sigma_r\|. \tag{49}$$

By (49) the process terminates: there exists some round $t \geqslant 0$ and during that round we reach a final state E_{fin} without variables and with an entire solution $(\alpha_{\text{fin}}, \mathrm{id}_C)$. Hence, the entire process defines a path in $\mathcal{F}$ which is labeled by some $h_1 \cdots h_t \in \mathrm{End}(C^*)$ such that $h_1 \cdots h_t(W_{\text{fin}}) = W_{\text{init}}$. We have $|W_{\text{init}}| = n$ and therefore $\|W_{\text{init}}\| \leqslant 30\kappa_1\delta n$. (Note that for large $n \to \infty$ the ratio $\frac{\|W_{\text{init}}\|}{30\kappa_1\delta n}$ tends to 0. For large n the initial size $\|W_{\text{init}}\|$ is much, much smaller than $30\kappa_1\delta n$. By Remark 11.11, for all rounds r with $0 \leqslant r \leqslant t$ we can state:

$$\|W_r\| \leqslant 30\kappa_1\delta n + 40\delta n \leqslant 2960\delta n. \tag{50}$$

We also need an estimation for the maximal weight of an equation in the middle of each round. Proposition 11.10 says we have to add at most $40\delta n$ with respect to the starting point of a round. Thus, the conclusion of (50) is therefore: whenever we see an equation $E = (W, B, \mathcal{X}, \theta, \mu)$ on the path from E_{init} to E_{fin} we have

$$\|W\| \leqslant 2960\delta n + 40\delta n \leqslant 3000\delta n. \tag{51}$$

Corollary 12.1. *Let $\kappa = 3000$ and let $\mathcal{B}$ the subautomaton of $\mathcal{F}$ which is defined defined as follows. The states of $\mathcal{B}$ are the extended equations $(W, B, \mathcal{X}, \theta, \mu)$ where*

$$\|W\| \leqslant \kappa\delta n.$$

Then $\mathcal{B}$ is a finite and complete subautomaton of $\mathcal{F}$. Let $\mathcal{A}_{\mathcal{S}}$ be the trimmed subautomaton of $\mathcal{B}$, then the NFA $\mathcal{A}_{\mathcal{S}}$ accepts a rational set of A-morphisms $L(\mathcal{A}_{\mathcal{S}}) \subseteq \mathrm{End}(C^*)$ satisfying the following conditions from Theorem 4.3

$$\mathrm{Sol}(\mathcal{S}) = \{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{\mathcal{S}})\}\,. \tag{52}$$

Moreover, $\mathrm{Sol}(\mathcal{S}) = \emptyset$ *if and only if* $L(\mathcal{A}_{\mathcal{S}}) = \emptyset$*; and* $|\mathrm{Sol}(\mathcal{S})| < \infty$ *if and only if* $\mathcal{A}_{\mathcal{S}}$ *doesn't contain any directed cycle.*

Proof. The automaton $\mathcal{B}$ is finite because first, the number of states is finite and second, if E is any state in $\mathcal{B}$, then there are only finitely many $E \xrightarrow{h} E'$ transitions in $\mathcal{F}$ where $E' \in \mathcal{B}$. Thus, the out-degree is finite for every state in $\mathcal{B}$. Since $\mathcal{B}$ is finite, $\mathcal{A}_{\mathcal{S}}$ is finite, too. The NFAs $\mathcal{A}_{\mathcal{S}}$ and $\mathcal{B}$ are both sound by Proposition 7.5. They are complete, this follows from Proposition 7.5, since $\kappa = 3000$ is large enough by (51). This shows

$$\mathrm{Sol}(\mathcal{S}) = \{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{\mathcal{S}})\}\,.$$

Finally, Proposition 8.3 implies that $\mathrm{Sol}(\mathcal{S}) = \emptyset$ if and only if $L(\mathcal{A}_{\mathcal{S}}) = \emptyset$ and that $|\mathrm{Sol}(\mathcal{S})| < \infty$ if and only if $\mathcal{A}_{\mathcal{S}}$ doesn't contain any directed cycle. □

12.1. The NSPACE algorithm to compute the trim NFA $\mathcal{A}_{\mathcal{S}}$.

The method is standard and is essentially the same as in [26, 11, 5]. Therefore we give a rough sketch only. The key is the upper bound in Corollary 12.1: it is enough to consider states $(W, B, \mathcal{X}, \theta, \mu)$ where $\|W\| \leqslant 3000\delta n \in \mathcal{O}(|H| \cdot n^2)$. This implies that the maximal length of an equation and the maximal number of H-visible letters is in $\mathcal{O}(|H| \cdot n^2) \subseteq \mathcal{O}(|H|\,\|\mathcal{S}\|^2)$. This in turn gives the upper bound $\mathcal{O}(|H|^2\|\mathcal{S}\|^2)$ on the alphabet C. It is also clear that we need at most $\mathcal{O}(\|\mathcal{S}\|)$ variables. To each symbol we have to attach its μ-value in the finite monoid N.

By Section 4.2 storing a μ-value costs $m(\mathcal{S})$ bits by (4). As a consequence we can specify a state E (and therefore a transition $E \xrightarrow{h} E'$) in $\mathcal{A}_{\mathcal{S}}$ with $\mathcal{O}(|H| \cdot \|\mathcal{S}\|^2 \cdot \log|A| \cdot m(\mathcal{S}) \cdot \log\|\mathcal{S}\|)$ bits.

Our algorithm must output all transitions $E \xrightarrow{h} E'$ which belong to $\mathcal{A}_{\mathcal{S}}$. Hence, we consider all candidates $E \xrightarrow{h} E'$ based on the upper bound of bits for their specification one after another in some order, say in some lexicographical order. The algorithm has to decide if it outputs the transition or whether it moves to the next candidate. Thus, when considering whether or not $E \xrightarrow{h} E'$ belongs to $\mathcal{A}_{\mathcal{S}}$, then the algorithm guesses a path of transitions from an initial state to the state E and a path of transitions from E' to a final state. If the guess is successful, then it outputs $E \xrightarrow{h} E'$ and it moves to the next candidate. If unsuccessful, then we apply again the theorem of Immerman-Szelepcsényi: $\mathsf{NSPACE}(|H|\,\|\mathcal{S}\|^2 \log|A|\, m(\mathcal{S}) \log\|\mathcal{S}\|)$ is closed under complementation. Hence, the algorithm "knows" whether or not $E \xrightarrow{h} E'$ belongs to $\mathcal{A}_{\mathcal{S}}$ before moving to the next candidate.

The proof of Theorem 4.3 is complete, and the first part of the paper is finished.

13. Part 2: The existential theory with rational constraints for virtually free groups

It was shown in [7, 35] that the existential theory with rational constraints in f.g. virtually free groups is decidable. Our main result (Theorem 14.2) provides an effective EDT0L description for the full set of satisfying assignments to a Boolean formula in free variables over equations and rational constraints. In order to make our statement precise we need some preparation.

13.1. **NFAs revisited.** Let M and M' be finitely generated monoids. In the application M and M' are fixed and not part of the input. Therefore we can define the size of an NFA over M (resp. M') which is, up to a constant, independent of the generating set.

We begin by choosing any finite generating set $\Sigma \subseteq M$. Then we specify an NFA for M as tuple $\mathcal{A} = (Q, \Sigma, \delta, \mathscr{I}, \mathscr{F})$ where the set of transitions δ is finite and satisfies $\delta \subseteq Q \times \Sigma^* \times Q$. Having this, a natural definition for the *input size of* $\mathcal{A}$ is

$$\|\mathcal{A}\|_{\mathrm{in},\Sigma} = |Q| + |\delta| + \sum_{(p,u,q)\in\delta} |u| \,. \tag{53}$$

The transitions in $\varphi(\mathcal{A})$ might be labeled by words of length greater than 1. However, this can be "repaired" easily by replacing a transition $(p, b_0 \cdots b_k, q)$ with $b_i \in \Gamma$ and $k > 0$ by a sequence of transitions

$$(p, b_0, p_1), \dots, (p_{k-1}, b_{k-1}, p_k), (p_k, b_k, q)$$

were $p_1, \dots, p_{k-1}$ are fresh states. The input size of the new automaton is at most twice as large as before. The exact size of an NFA $\mathcal{A}$ is of not important. We let

$$\|\mathcal{A}\|_{\mathrm{in}} = \Theta(\|\mathcal{A}\|_{\mathrm{in},\Sigma}). \tag{54}$$

This is well-defined, since if we move to another finite generating set Σ' for M, then we see $\|\mathcal{A}\|_{\mathrm{in},\Sigma'} \in \mathcal{O}(\|\mathcal{A}\|_{\mathrm{in},\Sigma})$. Thus, it is convenient to denote $\mathcal{A}$ simply as $\mathcal{A} = (Q, M, \delta, \mathscr{I}, \mathscr{F})$ because then the interpretation $L(\mathcal{A}) \subseteq M$ is encoded in the syntax. Still, we can use $\|\mathcal{A}\|_{\mathrm{in}}$ up to multiplicative constants.

Let $\varphi \colon M \to M'$ be a homomorphism to a monoid with a finite generating set Σ', then the NFA $\varphi(\mathcal{A})$ is defined as $(Q, M, \varphi(\delta), \mathscr{I}, \mathscr{F})$ where $\varphi(p, a, q) = (p, \varphi(a), q)$. For $s, t \in Q$ let $L(\mathcal{A}, s, t) = L(\mathcal{A}, M, \delta, \{s\}, \{t\})$. If $|\varphi(a)| \in \mathcal{O}(1)$ for all $a \in \Sigma$, then there is a result which is again independent of the choice of Σ and Σ':. We have

$$\|\varphi(\mathcal{A})\|_{\mathrm{in}} \in \Theta(\|\mathcal{A}\|_{\mathrm{in}}) \text{ and } \forall s, t \in Q : \varphi(L(\mathcal{A}, s, t)) = L(\varphi(\mathcal{A}), s, t)). \tag{55}$$

13.2. **Exponential expressions.** The ideas and results in this section are not new. The notion of *exponential expression* was proposed, for example, by Plandowski in [43]. For the application to $\mathrm{SL}(2, \mathbb{Z})$ exponential expressions are crucial to show a complexity within PSPACE. Intuitively it is more natural to represent strings by allowing exponents. For example, if u is a word, then it is more natural to write u^{100} rather than in plain form by repeating u a hundred times $uuuuuuuuuuuuuuuuuuuu \cdots$.

Exponential expressions (and plain exponential expressions) over an alphabet Σ and their sizes are defined inductively as follows.

(1) Every word $w \in \Sigma^*$ is a plain exponential expression of size $\|w\| = |w|$.
(2) Every plain exponential expression is an exponential expression.
(3) If E, E' are exponential expressions, then the concatenation EE' is an exponential expression of size $\|EE'\| = \|E\| + \|E'\|$. If E, E' are plain, then EE' is plain, too.
(4) If E is an exponential expression and $k \in \mathbb{N}$, then E^k is an exponential expression of size $\left\|E^k\right\| = 1 + \|E\| + \log k$.

Since Σ is equipped with an involution, we define for all $k \in \mathbb{Z}$ the expression E^{-k} as a synonym for $\overline{E}^k$; and we let E^0 denote the empty word 1. The size of the expression E^0 is still $\|E\| + 1$.

In the following we allow that an equation appearing in a Boolean formula Φ is written as $E = E'$ where E and E' are exponential expressions. We view E and E' as words Σ^* which have a special encoding in a compact form.

13.3. **The existential theory with constraints and expressions.** As above M denotes a finitely generated monoid with involution. We let $\Sigma \subseteq M$ be any finite symmetric set of generators: that is, $a \in \Sigma \implies \overline{a} \in \Sigma$. Let $\pi \colon \Sigma^* \to M$ be the canonical morphism which is induced by the inclusion $\Sigma \subseteq M$. By Ω we denote a countable set of variables such that $M \cap \Omega = \emptyset$. Without restriction we assume that Ω is a set with involution and $X \neq \overline{X}$ for all $X \in \Omega$. As usual, we let $\overline{g} = g^{-1}$ for group elements.

The existential theory of M with rational constraints and exponential expressions is defined with the help of Boolean formulae in free variables from Ω. As we did in Section 4.2, we obtain more accurate (and therefore better) complexity results if we define the size of a Boolean formula Φ as a pair $(\|\Phi\|_{\text{eq}}, \|\Phi\|_{\text{rat}})$. The parameter $\|\Phi\|_{\text{eq}}$ behaves as if all NFAs defining the rational constraints were of constant size. Thus, essentially, it adds up the sizes of the equations of the exponential expressions defining the equations. This is reflected by the index "eq". The parameter $\|\Phi\|_{\text{rat}}$ adds up the input sizes for the NFAs which define the rational constraints. This is reflected by the index "rat".

The formal definitions are as follows. Here we assume that every constraint $X \in L$ with $L \in \mathrm{Rat}(M)$ is given as $X \in \pi L(\mathcal{A})$ (resp. $X \in L(\mathcal{A})$) where $\mathcal{A}$ is an NFA as in Section 13.1. Exponential expressions were defined in Section 13.2.

(1) Every *atomic formula* is Boolean formula. The atomic formulae are:
- The constant $\perp$ (meaning "false")
  $\|\perp\|_{\text{eq}} = \|\perp\|_{\text{rat}} = 1$.
- Exponential expressions $E = E'$ over $(\Sigma \cup \Omega)^*$.
  $\|E = E'\|_{\text{eq}} = 1 + \|E\|_{\text{eq}} + \|E'\|_{\text{eq}}$ and $\|E = E'\|_{\text{rat}} = 0$.
- Constraints $X \in L(\mathcal{A})$.
  $\|X \in L(\mathcal{A})\|_{\text{eq}} = 1$ and $\|X \in L(\mathcal{A})\|_{\text{rat}} = \|\mathcal{A}\|_{\text{in}}$.

(2) If Φ, Ψ are Boolean formulae, then so are $(\Phi \vee \Psi)$, $(\Phi \wedge \Psi)$, and $(\neg\Phi)$, but we omit brackets when possible.
$\|\Phi \vee \Psi\|_\star = \|\Phi \wedge \Psi\|_\star = \|\Phi\|_\star + \|\Psi\|_\star$, and $\|\neg\Phi\|_\star = \|\Phi\|_\star$ for $\star \in \{\text{eq}, \text{rat}\}$.

Let Φ be a Boolean formula and $\sigma \colon \Omega \to M$ be a morphism (that is, a mapping respecting the involution). Then the truth value $\sigma(\Phi)$ is defined in the obvious way. If there exists some σ with $\sigma(\Phi) =$ true, then we say that Φ is *satisfiable*. We also say that σ is a *solution* if $\sigma(\Phi) =$ true because it solves the satisfiability problem. So, we do not distinguish between satisfying assignments and solutions. The *existential (first-order) theory with rational constraints* refers to the set of satisfiable Boolean formulae

$$\exists\mathrm{FOTh}(M, \mathrm{Rat}) = \{\Phi \mid \exists\, \sigma \colon \Omega \to M \text{ such that } \sigma(\Phi) = \text{ true}\}.$$

We are not only interested to *decide* $\exists\mathrm{FOTh}(M, \mathrm{Rat})$, what we aim for is an algorithm which produces on input a Boolean formula Φ an effective description of the *full solution set* $\mathrm{Sol}(\Phi, M)$. To define it properly we let $\mathcal{X}_\Phi$ be the set of variables X such that X or $\overline{X}$ appears in Φ. We let

$$\mathrm{Sol}(\Phi, M) = \{\sigma \colon \mathcal{X}_\Phi \to M \mid \sigma(\Phi) = \text{ true}\}. \tag{56}$$

Note that $\sigma \colon \Omega \to M$ satisfies Φ if and only if its restriction to $\mathcal{X}_\Phi$ satisfies Φ. It is also clear that every morphism $\sigma \colon \mathcal{X}_\Phi \to M$ satisfying Φ can be extended to a morphism $\sigma \colon \Omega \to M$ satisfying Φ. If the context of M is clear, we abbreviate $\mathrm{Sol}(\Phi) = \mathrm{Sol}(\Phi, M)$. Once we have chosen a presentation $\pi \colon S \to M$ where S is finite and π is onto, then we typically represent elements of M by words over S and a morphism $\sigma \colon \Omega \to M$ is defined via a mapping $\sigma \colon \Omega \to S^*$. Moreover, without restriction Ω comes with a linear order. If $\{X_1, \ldots, X_k\}$ is the subset of the first k

variables with $X_i \leqslant X_j$ for all $i \leqslant j$, then we let

$$\mathrm{Sol}_{S,k}(\Phi) = \{(\sigma(X_1), \dots, \sigma(X_k)) \in S^* \times \dots \times S^* \mid \exists \sigma : \pi\sigma(\Phi) = \text{ true}\}\,. \tag{57}$$

Clearly, to decide $\exists\mathrm{FOTh}(M, \mathrm{Rat})$ is the same as to decide on input Φ whether or not $\mathrm{Sol}(\Phi)$ is empty. Moreover, $\mathrm{Sol}(\Phi) = \emptyset \iff \mathrm{Sol}_{S,0}(\Phi) = \emptyset$. Note that either $\mathrm{Sol}_{S,0} = \emptyset$ or $\mathrm{Sol}_{S,0} = \{\emptyset\}$. We will see that $\mathrm{Sol}_{S,k}(\Phi)$ is an effective EDT0L relation for every k if M is a f.g. virtually free group.

When proving this result for virtually free groups we make various transformations on NFAs (which up to a constant factor don't change $\|\mathcal{A}\|_{\mathrm{in}}$) before, eventually, we switch to Boolean matrices.

13.4. **Removing exponential expressions in Φ.**

Exponential expressions in Boolean formulae as in (56) are used because they may reduce the size of Φ significantly. On the other hand, with the help of more variables we can transform Φ into a new formula Ψ where all equations are written in plain form as $U = V$. The transformation is not expensive; and it doesn't change the full solution set:

Proposition 13.1. *There is a deterministic algorithm working in linear space which takes as input a Boolean formula Φ using exponential expressions.*

The output is a formula Ψ having the following properties.

1. *Equations in Ψ appear in plain form as $U = V$. Hence, $\|UV\| = |UV|$.*
2. *$\|\Psi\|_{eq} \in \mathcal{O}(\|\Phi\|_{eq})$.*
3. *$\|\Psi\|_{rat} = \|\Phi\|_{rat}$.*
4. *$\mathcal{X}_\Phi \subseteq \mathcal{X}_\Psi$.*
5. *The restriction $(\sigma\colon \mathcal{X}_\Psi \to M) \mapsto (\sigma\colon \mathcal{X}_\Phi \to M)$ induces a bijection*

$$\mathrm{Sol}(\Psi) = \mathrm{Sol}(\Phi).$$

Proof. The method is standard: replace all exponential expressions by straight-line programs (SLPs), see for example [33, 34]. More precisely, as soon as an exponential expression $E = T^0$ with $e = 0$ appears, replace the expression E by the empty word 1. If an exponential expression T^1 appears in Φ, then replace every occurrence of T^1 simply by T. If an exponential expression $E = T^e$ with $e \geqslant 2$ appears in Φ, then define a fresh variable $[T, e]$. (This implicitly means to introduce $\overline{[T, e]} = [\overline{T}, e]$, too. We don't repeat this anymore.) Whenever a variable $[T, \ell]$ is introduced where $\ell \geqslant 2$, then we introduce another fresh variable $[T, \lfloor \ell/2 \rfloor]$, too. In particular, $[T, e]$ and $[T, 1]$ are introduced (but the condition $\ell \geqslant 2$ makes sure that $[T, 0]$ is never introduced). The total number of fresh variables $[T, \ell]$ introduced that way is bounded by $2(1 + \log e) \in \mathcal{O}(\log e)$.

After that step, replace all occurrences of E by $[T, e]$, if E was defined by $E = T^e$ in Φ; and for each fresh $[T, \ell]$ with $\ell \geqslant 2$ introduce a new plain equation

$$[T, \ell] = \begin{cases} [T, \lfloor \ell/2 \rfloor\,]\ [T, \lfloor \ell/2 \rfloor\,] & \text{if } \ell \text{ is even,} \\ [T, \lfloor \ell/2 \rfloor\,]\ [T, \lfloor \ell/2 \rfloor\,]\ [T, 1] & \text{otherwise.} \end{cases}$$

Moreover, introduce a single equation $[T, 1] = T$. The effect is that each occurrence of $E = T^e$, having size $\|E\| + \|T\| + 2 + \log(1 + e)$, is removed. The gain of $\|T\| + \log e$ is mitigated by $\mathcal{O}(\log e)$ new equations of constant size and one more equation $[T, 1] = T$ of size $\|T\| + 2$.

After that step replace Φ by the conjunction of Φ with the conjunction of the new equations. Continue until all equations are written in plain form. This defines the formula Ψ. Note that is not necessary to add any constraint on the fresh variables $[T, \ell]$. Therefore, $\|\Psi\|_{\mathrm{rat}} = \|\Phi\|_{\mathrm{rat}}$. The proposition follows. □

14. Virtually free groups

We restrict ourselves to the non-uniform complexity where the given virtually free group is not part of the input. The restriction allows us to ignore the way a virtually free group is given to us. For example whether the group is given by a context-free grammar for the word problem or whether it is given as a fundamental group of a finite graph of finite groups may result in uniform complexities which differ exponentially. We refer the interested reader to the arXiv version of [56] for more details. See also Remark 14.8.

In the following G denotes a fixed finitely generated virtually free group. Thus, there is a finitely generated free subgroup F such that $H = G/F$ is finite. Replacing F by the normal subgroup $\bigcap \left\{gFg^{-1} \;\middle|\; g \in G\right\}$ (which is of finite index in G) we can assume without restriction that F is normal and that H is a finite group. That is, we start with some surjective homomorphism $\gamma\colon G \to H$ where H is finite and the kernel $\ker(\gamma)$ is a f.g. free group. This yields a short exact sequence:

$$1 \to F \xrightarrow{\iota} G \xrightarrow{\gamma} H \to 1. \tag{58}$$

Choosing a generating set for F and a set of coset representatives from H, we obtain a generating set for G. We need generating sets which are closed under involution, so we are more specific. We use the following definition.

Definition 14.1. *Let G be given as in Equation (58). We say that a subset S of G is a* standard generating set *for G if the following conditions are satisfied.*

- *S can be written as a union $A_+ \cup A_- \cup H_+ \cup H_- \subseteq G$.*
- *A_+ is a basis for F, that is $F = F(A_+)$.*
- *$a \in A_+ \iff a^{-1} \in A_-$ for all $a \in A = A_+ \cup A_-$.*
- *γ induces a bijection between $H' = H_+ \cup \{1\}$ and H.*
- *$H_- = \left\{h \in G \;\middle|\; h^{-1} \in H_+\right\}$.*

Every standard generating set is closed under the involution with $\overline{b} = b^{-1} \in G$. The three set A_+, A_-, and H' are pairwise disjoint subsets of G. There is a bijection between H_+ and H_-, but perhaps $H_+ \cap H_- \neq \emptyset$.

Let $\pi_S\colon S^* \to G$ denote the canonical projection. We say that $\widehat{w} \in S^*$ is in *standard normal form* if we can write $\widehat{w} = uh$ where $u \in A^*$ is a freely reduced word (that is without factors $a\overline{a}$) and $h \in H'$. By $\mathrm{snf}_S(G)$ we denote the set of standard normal forms. For every $w \in S^*$ there is a unique $\mathrm{snf}_S(w) \in \mathrm{snf}_S(G)$ such that $w = \mathrm{snf}_S(w)$ in G. The set of freely reduced words over A becomes $A^* \cap \mathrm{snf}_S(G)$; and we let $\mathrm{snf}_A(G) = A^* \cap \mathrm{snf}_S(G)$. Hence,

$$\mathrm{snf}_A(G) = \{\mathrm{snf}_S(w) \mid w \in A^*\} = \{w \in A^* \mid w \text{ is freely reduced}\}\,. \tag{59}$$

Theorem 14.2. *Let G be a finitely generated virtually free group. Then with respect to any short exact sequence as in (58) there is a standard generating set S and an $\mathsf{NSPACE}(\|\Phi\|_{eq}^2 (\|\Phi\|_{rat}^2 + \log\|\Phi\|_{eq}))$ algorithm which performs the following task. It takes as input a Boolean formula Φ (according to Section 13.3) with $\mathcal{X}_\Phi = \left\{X_1, \overline{X}_1, \dots, X_k, \overline{X}_k\right\}$ such that X_i is the ith variable in some fixed chosen linear order on Ω. The output is an extended alphabet C of size $\mathcal{O}(\|\Phi\|_{eq}^2)$, letters $d_i \in C$ for all $1 \leqslant i \leqslant k$, and a trim NFA $\mathcal{A}_\Phi$ accepting a rational set of S-morphisms over C^* such that the EDT0L relation*

$$\{(h(d_1), \dots, h(d_k)) \in C^* \times \dots \times C^* \mid h \in L(\mathcal{A}_\Phi)\}$$

is equal to the full solution set in standard normal forms

$$\mathrm{Sol}_{S,k}(\Phi) = \left\{(\sigma(X_1), \dots, \sigma(X_k)) \in \mathrm{snf}_S(G)^k \;\middle|\; \pi_S\sigma(\Phi) = \textit{ true}\right\}.$$

Moreover, $\mathrm{Sol}(\Phi) = \emptyset$ if and only if $L(\mathcal{A}_\Phi) = \emptyset$; and $|\mathrm{Sol}(\Phi)| < \infty$ if and only if $\mathcal{A}_\Phi$ doesn't contain any directed cycle.

Remark 14.3. *In a simplified analysis using a single parameter, the natural choice is to define* $\|\Phi\| = \|\Phi\|_{eq} + \|\Phi\|_{rat}$*. This yields*

$$\mathsf{NSPACE}(\|\Phi\|_{eq}^2(\|\Phi\|_{rat}^2 + \log\|\Phi\|_{eq})) \subseteq \mathsf{NSPACE}(\|\Phi\|^4) \subseteq \mathsf{PSPACE}. \tag{60}$$

If $\|\Phi\|_{rat} \in \mathcal{O}(\sqrt{\log\|\Phi\|_{eq}})$*, then we have*

$$\mathsf{NSPACE}(\|\Phi\|_{eq}^2(\|\Phi\|_{rat}^2 + \log\|\Phi\|_{eq})) \subseteq \mathsf{NSPACE}(\|\Phi\|^2\log\|\Phi\|). \tag{61}$$

Remark 14.4. *Let us comment why we consider only a subset of the first k variables of* $\mathcal{X}_+$ *rather than all variables. The reason is that during the proof we manipulate* Φ *in various ways including some which introduce fresh variables. But these new variables are just auxiliary symbols, and we make sure that they don't enlarge the full solution set. If we introduce fresh variables, then we put them in the linear order behind the first k variables. Therefore, there is no risk to denote* $\mathrm{Sol}(\Phi)$ *as* $\mathrm{Sol}_{S,k}(\Phi)$*.*

14.1. **Proof of Theorem 14.2, Phase 1.** Using Proposition 13.1 we may assume that all equations in Φ are written in plain form as $U = V$ where $U, V \in (S \cup \mathcal{X})^*$. In the following we introduce many fresh variables into $\mathcal{X}$. The enlarged set is still called $\mathcal{X}$. Moreover, we choose a subset of *positive* variables $\mathcal{X}_+$ such that $\{X_1, \dots, X_k\} \subseteq \mathcal{X}_+$, $\mathcal{X} = \mathcal{X}_+ \cup \{\overline{X} \mid X \in \mathcal{X}_+\}$, and $X \in \mathcal{X}_+ \iff \overline{X} \notin \mathcal{X}_+$.

Having this, we push all negations to the atomic formulae using De Morgan's law. This increase the size at most by the number of atomic formulae. For each inequality $\neg(U = V)$ we introduce a fresh variable X and then we replace $\neg(U = V)$ by the conjunction $U = VX \wedge \neg(X \in \{1\})$. This increases the size by the number of inequalities since the singleton $\{1\}$ is accepted by a two-state NFA. Thus, without restriction Φ doesn't contain any negation and only three types of atomic formulae:

$$U = V, \quad X \in \pi_S(L(\mathcal{A})), \quad \text{and } X \notin \pi_S(L(\mathcal{A})) \text{ where } U, V \in S^* \text{ and } \ X \in \mathcal{X}_+.$$

Here, denotes an NFA of the form $\mathcal{A} = (Q, S, \delta, \mathscr{I}, \mathscr{F})$ with $\delta \subseteq Q \times S \times Q$. We may assume because for every NFA $\mathcal{A}$ there is another NFA $\overline{\mathcal{A}}$ of the same size such that $L(\overline{\mathcal{A}}) = \overline{L(\mathcal{A})}$ (the complement of $L(\mathcal{A})$). We also write $X \in L(\mathcal{A})$ or $X \notin L(\mathcal{A})$ because $S \subseteq G$ and therefore we can view $L(\mathcal{A})$ directly as a rational subset of G.

Lemma 14.5. *Let* $\gamma\colon G \to H$ *be as above. In particular,* $F = \gamma^{-1}(H)$ *is free and* H *is finite. It is enough to prove Theorem 14.2 under the following assumptions about the input formula* Φ*.*

- Φ *implies* $\bigwedge \{X \in F \mid X \in \mathcal{X}_+\}$. *(Note that the syntax* $X \in F$ *makes sense since the f.g. free group* F *is a rational subset in* G*.)*
- *If an NFA* $\mathcal{A}$ *appears in* Φ*, then* $L(\mathcal{A}) \subseteq F$ *where* $\mathcal{A}$ *is an NFA over* G *and the transitions are labeled with an arbitrary, but fixed, finite set of generators of* G*.*
- Φ *is a conjunction where each atomic formula is either an equation in plain form* $U = V$ *with* $UV \in (S \cup \mathcal{X})^*$ *or* $X \in L(\mathcal{A})$ *or* $X \notin L(\mathcal{A})$*.*

Remark 14.6. *Assume that* Φ *satisfies the assumptions of Lemma 14.5. Then Theorem 14.2 implies that there is standard set of generators* S *containing a basis* A_+ *of* F *such that*

$$\mathrm{Sol}_{S,k}(\Phi) = \left\{(\sigma(X_1), \dots, \sigma(X_k)) \in \mathrm{snf}_A(G)^k \;\middle|\; \pi_S\sigma(\Phi) = \ \mathit{true}\right\}.$$

In particular, the full solution set $\mathrm{Sol}_{S,k}(\Phi)$ *is an EDT0L relation over freely reduced words of* A*.*

The proof of Lemma 14.5 is based on two closure properties: 1. finite unions of EDT0L (resp. rational) languages in a monoid M are EDT0L (resp. rational); and 2. if $L \subseteq M$ is an EDT0L (resp. rational) language and $m \in M$, then Lm is EDT0L (resp. rational). The analogous statements hold for EDT0L relations.

Proof of Lemma 14.5. The difficult part is to show the first and third item because we have to respect the given space bounds. The second item is very easy to show, and we prove it "on the fly" when showing the first item.

Let Φ be any input formula for Theorem 14.2. We wish to add the constraint $X \in F$ for all variables. This requires the introduction of fresh variables. More precisely, for each $X \in \mathcal{X}_+$ and $g \in H'$ we introduce a new variable X_g with $\overline{X_g} = \overline{X}_g$; and we construct an NFA $\mathcal{A}_g$ such that $L(\mathcal{A}_g) = L(\mathcal{A})\overline{g}$ for each $\mathcal{A}$. The NFA $\mathcal{A}_g$ is obtained by adding a single new final state and new transitions from the former final states to the new single one, all of them are labeled by the letter $\overline{g}$. The size of $\mathcal{A}_g$ increases by a constant. Moreover, each function $\eta\colon \mathcal{X}_+ \to H$ defines a new formula Φ'_η over the variables $X_{\eta(X)}$ as follows: every occurrence of X (resp. $\overline{X}$) inside an equation is replaced by $X_g\overline{g}$ (resp. $g\overline{X}_g$) where $g = \eta(X)$. The length of each equation is at most doubled. Every constraint $X \in L(\mathcal{A})$ (resp. $X \notin L(\mathcal{A})$) is replaced by the $X_g \in L(\mathcal{A}_g)$ (resp. $X_g \notin L(\mathcal{A}_g)$). Recall that $L(\mathcal{A}_g) = L(\mathcal{A})\overline{g}$. Let Φ'_η denote the result of that transformation. Then we let

$$\Phi_\eta = \Phi'_\eta \wedge \bigwedge \left\{ X_{\eta(X)} \in F \;\middle|\; X \in \mathcal{X}_+ \right\}. \tag{62}$$

Note that a constraint $X_g \in F$ is the same as $\gamma(X_g) = 1$. Therefore we can use H as a recognizing finite monoid for all X_g. Since H is of constant size, the size of Φ_η is in $\mathcal{O}(\|\Phi\|_{\text{eq}}, \|\Phi\|_{\text{rat}})$. All variables in Φ_η are of the form $X_{\eta(X)}$ or $\overline{X}_{\eta(X)}$. The old variables $X \in \mathcal{X}$ are still present but not used in any Φ_η. Therefore, inside each Φ_η we rename all $X_{\eta(X)}$ by X. After that the variables X_g are superfluous: we remove them from $\mathcal{X}$. Thus, each Φ_η uses the same set of $\mathcal{V}_+$ as Φ did.

Each formula Φ_η is written again in disjunctive normal form $\Phi_\eta = \bigvee \{\Phi_{\eta,j} \mid j \in I_\eta\}$ where each index set I_η has again (at most) exponential in the size of Φ.

Having this, we see that Φ is equivalent to the following disjunction

$$\widetilde{\Phi} = \bigvee \left\{ \Phi_{\eta,j} \;\middle|\; \eta \in H^{\mathcal{X}_+} \wedge j \in I_\eta \right\}. \tag{63}$$

Note that Φ and $\widetilde{\Phi}$ use the same set $\mathcal{X}_+$ of positive variables. It is also clear how to transform a solution σ for Φ into a solution $\widetilde{\sigma}$ for $\widetilde{\Phi}$ and vice versa: if σ solves Φ, then $\widetilde{\sigma}(X_g) = \sigma(X)g$ solves $\widetilde{\Phi}$ and if $\widetilde{\sigma}$ solves $\widetilde{\Phi}$, then $\sigma(X) = \widetilde{\sigma}(X_g)\overline{g}$ solves Φ.

Since $|H|$ is a constant it is easy to see that the number of disjunctions in Equation (63) is (at most) exponential in the size of Φ. But it can also happen that the size of $\widetilde{\Phi}$ is exponential in the size of Φ, so in general we have no way to store $\widetilde{\Phi}$ within the given space bound. What we do instead is to construct NFAs $\mathcal{A}_{\eta,j}$ for each $\Phi_{\eta,j}$, one after another, such that $L(\mathcal{A}_{\eta,j})$ defines the EDT0L relation of the full solution set for $\Phi_{\eta,j}$.

More precisely, suppose we have shown Theorem 14.2 for each $\Phi_{\eta,j}$ which is a conjunction of constraints and equations. Then indeed, for all (η, j), one after another, we can output some NFA $\mathcal{A}_{\eta,j}$ where the transitions are labeled by endomorphisms over (the same) extended alphabet C such that

$$\mathrm{Sol}_{S,k}(\Phi_{\eta,j}) = \{(h(d_1), \dots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{\eta,j})\}. \tag{64}$$

We can also assume that all these NFAs use exactly the same set of distinguished letters $\{d_1, \dots, d_k\}$. As an output of the overall algorithm we obtain the disjoint union over all these NFAs $\mathcal{A}_{\eta,j}$. Without restriction $H' \cup H_- \subseteq C$, but the elements of $H' \cup H_-$ are not used in any NFA so far. Moreover, for each $d_i, \overline{d_i}$ we may assume that there are letters $c_i, \overline{c_i}$, again still not used by any $\mathcal{A}_{\eta,j}$. We add one more new state and connect this new state with all final states in $\mathcal{A}_{\eta,j}$ via a single transition labeled by the endomorphism $h \in \mathrm{End}(C^*)$ which is defined by $h(c_i) = d_i g$ where d_i corresponds to the variable $X_i \in \mathcal{X}_+$ and $g = \eta(X_i) \in H'$. The new state becomes the single final state of the "union" automaton.

We conclude that it is enough to show Theorem 14.2 for each $\Phi_{\eta,j}$. Since $\Phi_{\eta,j}$ satisfies properties as required by Lemma 14.5, the lemma follows. □

14.2. **Proof of Theorem 14.2, Phase 2.**

Embedding into a semi-direct product. Let E be a finite set with involution. Then $\mathbb{F}(E)$ denotes the group $\mathbb{F}(E) = E^*/\{e\overline{e} = 1 \mid e \in E\}$. If the involution on E is without fixed points, then we can write $E = E_+ \cup E_-$ such that $e \in E_+ \iff \overline{e} \in E_-$; and the inclusion E_+ into E induces an isomorphism between the free group $F(E_+)$ with basis E_+ and $\mathbb{F}(E)$. The group $\mathbb{F}(E)$ is called *specular* in [3], which means it is the free product of a free group with groups of order two.

In the following we use that G is the fundamental group of a finite graph of finite groups [29], which enables us to reduce questions about equations with rational constraints in G to questions about twisted word equations with rational constraints.

Suppose that the group H acts on E via a morphism $H \to \mathrm{Aut}(E)$. Thus, for $(f, e) \in H \times E$ we have $f(e) \in E$ and $f(\overline{e}) = \overline{f(e)}$. We have $\mathrm{Aut}(E) = \mathrm{Aut}(E^*) \subseteq \mathrm{Aut}(\mathbb{F}(E))$ and the action of H on E^* and $\mathbb{F}(E)$ defines two different (but related) semi-direct products $E^* \rtimes H$ and $\mathbb{F}(E) \rtimes H$. The elements of $E^* \rtimes H$ (resp. $\mathbb{F}(E) \rtimes H$) are the pairs $(u, f) \in E^* \times H$ (resp. $\mathbb{F}(E) \times H$) and the multiplication is defined by

$$(u, f) \cdot (v, g) = (uf(v), fg).$$

The semi-direct product $E^* \rtimes H$ is a monoid with involution by

$$\overline{(u, f)} = (f^{-1}(\overline{u}), f^{-1}).$$

It is also clear that $(u, f)^{-1} = (f^{-1}(u^{-1}), f^{-1})$ in the group $\mathbb{F}(E) \rtimes H$.

The free monoid E^* embeds into $E^* \rtimes H$ via $e \mapsto (e, 1)$ and the group H embeds into $E^* \rtimes H$ via $f \mapsto (1, f)$. Having this, we obtain:

$$\begin{aligned}\mathbb{F}(E) \rtimes H &= (E^*/\{e\overline{e} = 1 \mid e \in E\}) \rtimes H \\ &= E^* \times H/\{(e, 1)(\overline{e}, 1) = (1, 1) \mid e \in E\}.\end{aligned}$$

Since we identify E with $E \times \{1\}$ and H with $\{1\} \times H$, we can write:

$$\forall a \in E, g \in H : ga\overline{g} = (1, g)(a, 1)(1, \overline{g}) = (g(a), 1) = g(a). \tag{65}$$

Thus, $gx\overline{g} = g(x)$ for $g \in H$, $x \in E^*$, and $x \in \mathbb{F}(E)$. Let $\Gamma = E \cup H$ and $H \cap E^* = \{1\}$. Thus, 1 the identity element in H is identified with the empty word in E^*. It also appears as a letter in Γ. The interpretation $e \in E$ as $(e, 1)$ and $f \in H$ as $(1, f)$ yields canonical surjective morphisms

$$\pi_\Gamma \colon \Gamma^* \to E^* \rtimes H \to \mathbb{F}(E) \rtimes H. \tag{66}$$

Our proof of Theorem 14.2 relies on Proposition 14.7.

Proposition 14.7 ([13], Sec. 2.4.5)**.** *Let G be a finitely generated virtually free group and $\gamma \colon G \to H$ be a homomorphism onto a finite group H such that the kernel $F = \ker(\gamma)$ is free. Then G embeds into a semi-direct product of the form $\mathbb{F}(E) \rtimes H$; and we can construct an injective homomorphism $\varphi \colon G \to \mathbb{F}(E) \rtimes H$ and a partition $E = A \cup T$ into two subalphabets such that $F = \{x \in G \mid \varphi(x) \in \mathbb{F}(E)\}$ is isomorphic to $\mathbb{F}(A)$. Moreover, using that isomorphism, we can embed $\mathbb{F}(A)$ into G such that $\varphi(a) \in T^*aT^*$ such that $\varphi(a)$ is freely reduced in E^*. The embedding of G into $\mathbb{F}(E) \rtimes H$ is also depicted in Figure 9.*

$$\begin{tikzcd}
1 \arrow[r] & \mathbb{F}(A) \arrow[r] \arrow[d, "\varphi_A"] & G \arrow[r, "\gamma"] \arrow[d, "\varphi"] & H \arrow[r] \arrow[d, "\mathrm{id}_H"] & 1 \\
1 \arrow[r] & \mathbb{F}(E) \arrow[r] & \mathbb{F}(E) \rtimes H \arrow[r] & H \arrow[r] & 1
\end{tikzcd}$$

FIGURE 9. Embedding of G with $E = A \cup T$ and $\varphi_A(a) = \varphi(a) \in T^*aT^*$.

Proof. The first assertion is the nontrivial direction in [13, Cor. 2.4.23]. The corollary says that $F = \{x \in G \mid \varphi(x) \in \mathbb{F}(E)\}$ is a free factor of $\mathbb{F}(E)$. This means that $\mathbb{F}(E)$ is a free product of $\mathbb{F}(A)$ with $\mathbb{F}(T)$. The additional property $\varphi(a) \in T^*AT^*$ for all $a \in A$ is a special case of [13, Prop. 2.4.22]. □

Remark 14.8. *As we deal here with non-uniform complexity, we content ourselves to know that the embedding $\varphi\colon G \to \mathbb{F}(E) \rtimes H$ can be effectively computed and therefore we can treat $|E|$ as a constant. But in fact the proof of Lemma 30 in the arXiv version of* [56] *shows*

$$|E| \leqslant (|A| + 2\,|H|) \cdot |H|\,. \tag{67}$$

Thus, if G is given to us as the fundamental group of a finite graph of finite groups, then the interested reader could derive uniform complexity bounds from the material presented here.

Corollary 14.9. *We use the same notation as in Proposition 14.7. Define a morphism ψ from $\mathbb{F}(E)$ onto $\mathbb{F}(A)$ by $\psi(a) = a$ and $\psi(t) = 1$ for $a \in A$ and $t \in E \setminus A$. Then ψ maps freely reduced words $\widehat{w} \in \varphi(A^*)$ to freely reduced words in A^*.*

Proof. The subgroup $\varphi(\mathbb{F}(A))$ of $\mathbb{F}(E)$ is generated by words of the form $\varphi(v)$ where $v \in A^*$ is freely reduced over A. Thus, we can write every element in $w \in \varphi(\mathbb{F}(A))$ as a word

$$w = \varphi(a_1) \cdots \varphi(a_m) \in T^*a_1T^* \cdots T^*a_mT^*$$

such that $a_i \neq \overline{a_{i+1}}$ for $1 \leqslant i < m$. Now, every freely reduced word $\widehat{w}$ can be obtained from some word w as above by cancellation of factors $e\overline{e}$. Since $a_i \neq \overline{a_{i+1}}$ for $1 \leqslant i < m$, we obtain

$$\widehat{w} \in \varphi(a_1) \cdots \varphi(a_m) \in T^*a_1T^* \cdots T^*a_mT^*.$$

In particular, $\psi(\widehat{w}) = a_1 \cdots a_m$, and $a_1 \cdots a_m$ is freely reduced by definition. □

Remark 14.10. *Let us give a few more comments how Proposition 14.7 and Corollary 14.9 are shown in* [13]. *Since G is a fundamental group of a finite graph of finite groups, it acts on its Bass-Serre tree $\mathcal{T}$ without edge inversion* [57]. *As the notation suggests, $\mathcal{T}$ is indeed a tree: a connected acyclic undirected graph. The same is true for the free subgroup $F = \ker(\gamma)$ of G: it acts on $\mathcal{T}$ as a graph automorphism without edge inversion. It follows that F has trivial intersection with all vertex groups because the vertex groups are finite and embed into G, see again* [57]. *So, if the intersection was not trivial, the G would have a finite nontrivial subgroup, but free groups are torsion free. Thus, F acts on $\mathcal{G}$ without vertex stabilizers and without without edge inversion.*

Now, let $\mathcal{G}$ be the quotient graph $\mathcal{G} = F \setminus \mathcal{T}$. The finite group H acts on $\mathcal{G}$: it permutes the edges and vertices of $\mathcal{G}$ by respecting the incidence relation. Moreover, F appears as the fundamental group of the finite and connected simplicial graph $\mathcal{G}$.

This can be viewed as the main structure theorem about groups acting on trees, [57, Thm. 13]*. That is, we can write $F = \pi_1(\mathcal{G})$. The point is that we always have two views on fundamental groups of a simplicial graph. The first view is to choose a base point $\star$ and we write $\pi_1(\mathcal{G}) = \pi_1(\mathcal{G}, \star)$ where $\pi_1(\mathcal{G}, \star)$ is a set of paths from $\star$ to $\star$. The second view is to choose a spanning tree T of $\mathcal{G}$ and we realize $\pi_1(\mathcal{G})$ as $\pi_1(\mathcal{G}) = \pi_1(\mathcal{G}, T)$. That is $\pi_1(\mathcal{G}) = \pi_1(\mathcal{G}, T) = \mathbb{F}(E)/\{t = 1 \mid t \in T\}$. The isomorphism between $\pi_1(\mathcal{G}, \star)$ and $\pi_1(\mathcal{G}, T)$ is induced by the inclusion of $\pi_1(\mathcal{G}, \star)$ into $\mathbb{F}(E)$ followed by the projection $\mathbb{F}(E)$ onto the quotient $\pi_1(\mathcal{G}, T) = \mathbb{F}(E)/\{t = 1 \mid t \in T\}$.*

Let E be the set of edges in $\mathcal{G}$. It is (in the sense of [57]*) a finite alphabet: a finite set with involution without fixed points. Thus, we can view E as a disjoint union $E_+ \cup E_-$ with $e \in E_+ \iff \overline{e} \in E_-$. As a set we identify $\mathbb{F}(E) = F(E_+) = E^*/\{e\overline{e} = 1 \mid e \in E\}$ with the regular set of reduced words in E^*. Recall that a word is reduced if and only if no factor $e\overline{e}$ for $e \in E$ appears. Since H acts on the graph $\mathcal{G}$, each $g \in H$ acts on E^* via a length preserving automorphism which respects the involution. Hence, w is reduced if and only if $g(w)$ is reduced.*

14.3. **Proof of Theorem 14.2, Phase 3.**

Transformation of Φ to Ψ. Let Φ be the input formula for showing Theorem 14.2. By Lemma 14.5 we may assume that $\Phi = \Phi_{\eta,j}$ where $\Phi_{\eta,j}$ appears in Equation (63). Thus, Φ is given as a single conjunction of a special form where every variable is bounded by a constraint $X \in \mathbb{F}(A)$. It follows that the choice of H' for the standard generating set doesn't effect $\mathrm{Sol}_{S,k}(\Phi)$ from this point on. Therefore, we can write

$$\pi_A \mathrm{Sol}_{A,k}(\Phi) = \pi_A \mathrm{Sol}_{S,k}(\Phi) = \pi_S \mathrm{Sol}_{S,k}(\Phi) \tag{68}$$

Next, we use the embedding of G into the semi-direct product $\mathbb{F}(E) \rtimes H$ as given by Proposition 14.7 and Figure 9. We are going to transform Φ into a formula Ψ over $\mathbb{F}(E) \rtimes H$ such that the inclusion of G into the semi-direct product defines a bijection between $\mathrm{Sol}(\Phi)$ and $\mathrm{Sol}(\Psi)$.

We construct Ψ according to the following steps.

(1) We extend the embedding $\varphi\colon G \to \mathbb{F}(E) \rtimes H$ to an embedding $\varphi \star \mathrm{id}_{\mathcal{X}}\colon G \star \mathcal{X}^* \to (\mathbb{F}(E) \rtimes H) \star \mathcal{X}^*$ and we replace every equation $U = V$ in Φ by $\varphi \star \mathrm{id}_{\mathcal{X}}(U) = \varphi \star \mathrm{id}_{\mathcal{X}}(V)$. Identifying E, H, and $\mathcal{X}$ with subsets of $(\mathbb{F}(E) \rtimes H) \star \mathcal{X}^*$, we see that $E \cup H \cup \mathcal{X}$ generates the group $(\mathbb{F}(E) \rtimes H) \star \mathcal{X}^*$. Hence, every equation $\varphi \star \mathrm{id}_{\mathcal{X}}(U) = \varphi \star \mathrm{id}_{\mathcal{X}}(V)$ can be written as a plain equation over the alphabet $E \cup H \cup \mathcal{X}$. As we have defined $\Gamma = E \times \{1\} \cup \{1\} \times H = E \cup H$, we have $E \cup H \cup \mathcal{X} = \Gamma \cup \mathcal{X}$.

(2) We replace every $\mathcal{A}$ which appears in Φ by $\mathcal{A}_1 = \varphi(\mathcal{A})$. That is $L(\mathcal{A}_1) = \varphi(L(\mathcal{A}))$. By assumption we have $L(\mathcal{A}_1) \subseteq \mathbb{F}(E)$. Hence, $L(\mathcal{A}_1)$ is a rational subset of the free group $\mathbb{F}(E)$.

    Note that $\|\mathcal{A}_1\| \in \mathcal{O}(\|\mathcal{A}\|)$. Without restriction, we may assume that the transitions in $\mathcal{A}_1$ are labeled by elements from Γ. The property $\|\mathcal{A}_1\| \in \mathcal{O}(\|\mathcal{A}\|)$ is not effected by that assumption. Let Ψ_1 be the intermediate formula. It is clear that φ induces a bijection between $\mathrm{Sol}(\Phi)$ and $\mathrm{Sol}(\Psi_1)$.

(3) We transform each $\mathcal{A}_1$ appearing in Ψ_1 into an NFA $\mathcal{B}$ such that first, $L(\mathcal{A}_1) = L(\mathcal{B}) \subseteq \mathbb{F}(E)$ and second, the transitions $\mathcal{B}$ use labels from $E \cup \{1\}$, and third $\|L(\mathcal{B})\| \in \mathcal{O}(\|L(\mathcal{A}_1)\|)$. This well known by [55, 58], but not completely obvious. In Lemma 14.11 we give a slightly simplified proof for the special situation of semi-direct products.

    Let Ψ be the corresponding formula. Since $L(\mathcal{A}_1) = L(\mathcal{B})$, we have $\mathrm{Sol}(\Psi) = \mathrm{Sol}(\Psi_1)$.

The construction of Ψ is finished. We have

$$\varphi(\mathrm{Sol}(\Phi)) = \mathrm{Sol}(\Psi) \text{ and } (\|\Psi\|_{\mathrm{eq}}, \|\Psi\|_{\mathrm{rat}}) \in \mathcal{O}(\|\Phi\|_{\mathrm{eq}}, \|\Phi\|_{\mathrm{rat}}). \tag{69}$$

Lemma 14.11 ([55, 58]). *Let $\mathcal{A} = (Q, \Gamma, \delta, \mathscr{I}, \mathscr{F})$ be an NFA with the property $L(\mathcal{A}) \subseteq \mathbb{F}(E)$. Then there is an NFA $\mathcal{B}$ such that $L(\mathcal{A}) = L(\mathcal{B})$ where the transitions $\mathcal{B}$ use labels from $E \cup \{1\}$, and $\|L(\mathcal{B})\|_{in} \in \mathcal{O}(\|L(\mathcal{A})\|_{in})$.*

Proof. In the beginning we let δ be any finite subset of $Q \times \Gamma^* \times Q$. By Section 13.1 and perhaps by doubling the size of $\mathcal{A}$, we may assume that $\delta \subseteq Q \times (\Gamma \cup \{1\}) \times Q$. Since $\Gamma = E \cup H$ and $1 \in H$ we may assume that $\delta \subseteq Q \times H(E \cup \{1\}) \times Q$. Thus, the label of every transition is either an element from H or from $E \cup \{1\}$ or a product ha where $h \in H$ and $a \in E$. Moreover, we may assume that $\mathcal{A}$ is trim. In particular, if we reach a state p when reading a word u from an initial state, then there is a word v such that $uv \in L(\mathcal{A})$. Now, $L(\mathcal{A}) \subseteq \mathbb{F}(E)$. We let $\gamma(p) = \gamma(u)$. This is well-defined as $\gamma(u)\gamma(v)^{-1} = 1$.

For every state p with $\gamma(p) = g$ we introduce exactly one more state $[p, g]$ and transitions $p \xrightarrow{\overline{g}} [p, g]$ and $[p, g] \xrightarrow{g} p$. This does not change the language accepted, and the NFA is still trim with $\gamma([p, g]) = 1$. For each outgoing transition $p \xrightarrow{ha} q$ with $h \in H$ and $a \in E \cup \{1\}$ we have $\gamma(q) = f = gh$; and there is some $b \in E \cup \{1\}$ such that $bf = fa$ in G and hence, we add a transition $[p, g] \xrightarrow{b} [q, f]$ as depicted in Figure 10.

$$\begin{tikzcd}
p \arrow[r, "ha"] \arrow[d, "\overline{g}"] & q \\
{[p,g]} \arrow[r, "b"] & {[q,f]} \arrow[u, "f"']
\end{tikzcd}$$

FIGURE 10. The equations $bf = af$ and $f = gh$ imply $\overline{g}bf = ha$.

This doesn't change the language accepted as $\overline{g}bf = ha$ in G. The larger NFA still accepts L, but the crucial point is that for $h_1a_1 \cdots h_ka_k \in L(A)$ we can accept the same element in G by reading just labels from $E \cup \{1\}$. This is easy to see by induction on k. Now, we remove all original states (they are no longer needed) and make $[p, 1]$ initial (resp. final) if and only if p was initial (resp. final), to obtain the NFA $\mathcal{B}$. By construction, we have $\|\mathcal{B}\|_{\text{in},\Gamma} \leqslant 2\,\|\mathcal{A}\|_{\text{in},\Gamma}$. This implies $\|\mathcal{B}\|_{\text{in}} \in \mathcal{O}(\|\mathcal{A}\|_{\text{in}})$. Recall that $\|\mathcal{A}\|_{\text{in}}$ is well-defined up to a multiplicative constant only by Equation (54). This makes $\|\mathcal{A}\|_{\text{in}}$ independent of the choice of a finite generating set. □

14.4. **Proof of Theorem 14.2, Phase 4.**

From Ψ to $\Psi_{\mathbf{Ben}}$: applying the techniques of Benois. The transformation in this subsection doesn't effect the equations in Ψ. We only change the NFAs $\mathcal{B}$ such that they accept with every word $w \in E^*$ also the word $\widehat{w}$ which is obtained by canceling all factors $e\overline{e}$. Nevertheless the rational subset $\pi_E(L(\mathcal{B})) \subseteq \mathbb{F}(E)$ will not change. The techniques for the transformation is well known by the work of Michèle Benois [2]. Therefore we call the new formula Ψ_{Ben}. We will see that $\pi_E(\text{Sol}_{E,k}(\Psi)) = \pi_E(\text{Sol}_{E,k}(\Psi_{\text{Ben}}))$. For convenience of the reader, we explain the transformation in detail. We use notation from string rewriting.

For $u, v \in E^*$ we write $u \Rightarrow v$ if $u = pq$ and $v = pe\overline{e}q$ for some $p, q \in E^*$ and $e \in E$. By $\overset{*}{\Rightarrow}$ we mean the reflexive and transitive closure of $\Rightarrow$. Clearly, $u \overset{*}{\Rightarrow} v$ implies $\pi_E(u) = \pi_E(v)$. Moreover, $\pi_E(u) = \pi_E(v)$ implies $u \overset{*}{\Rightarrow} w \overset{*}{\Leftarrow} v$ for some $w \in E^*$.

By $\mathbb{F}$ we denote the set of freely reduced words over E; and we identify $\mathbb{F}(E)$ with the regular set $\mathbb{F} \subseteq E^*$. These a words without any factor $e\overline{e}$ where $e \in E$ or, equivalently, the set of words u such that $u \overset{*}{\Rightarrow} v$ implies $u = v$. The identification as sets is possible because π_E yields a bijection from $\mathbb{F}$ onto $\mathbb{F}(E)$.

The formula Ψ uses NFAs $\mathcal{B}$ where the transitions are labeled by letters from E or by the empty word 1, see Section 14.3. The interpretation so far is that $L(\mathcal{B})$ denotes a rational subset in $\mathbb{F}(E)$. Now, we switch the viewpoint: $L(\mathcal{B})$ denotes a regular subset in E^*; and we replace all constraints $X \in \pi_E L(\mathcal{B})$ (resp. $X \notin \pi_E L(\mathcal{B})$) by $X \in L(\mathcal{B})$ (resp. $X \notin L(\mathcal{B})$). This is nothing but a change of notation, so we call the new formula still Ψ. However, we now on we consider the full solution set $\mathrm{Sol}_{E,k}(\Psi)$ as a relation over E^*. Thus, a solution σ is given as a morphism $\sigma\colon \mathcal{X} \to E^*$. Recall that the "actual" solution over the group $\mathbb{F}(E) \rtimes H$ is therefore given by $\pi_E\sigma\colon \mathcal{X} \to \mathbb{F}(E)$.

Lemma 14.12 ([2])**.** *Let $\mathcal{B} = (Q, E, \delta; \mathscr{I}, \mathscr{F})$ an NFA which appears in Ψ. Thus, $\delta \subseteq Q \times (E \cup \{1\}) \times Q$ and $L(\mathcal{B}) \subseteq E^*$. Then we can transform $\mathcal{B}$ into an NFA $\mathcal{B}' = (Q', E, \delta'; \mathscr{I}', \mathscr{F}')$ such that $|Q'| = |Q|$, $\delta' \subseteq Q' \times (E \cup \{1\}) \times Q'$, and*

$$\pi_E(L(\mathcal{B}') \cap \mathbb{F}) = \pi_E(L(\mathcal{B}')) = \pi_E(L(\mathcal{B})). \tag{70}$$

Proof. Let $\mathcal{B} = (Q, E, \delta, \mathscr{I}, \mathscr{F})$ an NFA over E^* where $\delta \subseteq Q \times (E \cup \{1\}) \times Q$. We run the following while-loop.

While there there are a letter $e \in E$ and states $s, t \in Q$ such that $(s, 1, t) \notin \delta$ but $e\overline{e} \in L(\mathcal{B}, s, t)$ enlarge δ by the ε-transition $(s, 1, t)$.

The while-loop terminates after at most $|Q|^2$ rounds with the desired NFA $\mathcal{B}'$. The number of states is same as before.

The inclusion $L(\mathcal{B}') \subseteq \left\{ v \in E^* \;\middle|\; u \overset{*}{\Rightarrow} v \wedge u \in L(\mathcal{B}) \right\}$ is trivial. The converse follows by induction on the length of u. Moreover, for each $u \in E^*$ there is a (unique) $\widehat{u} \in \mathbb{F}$ such that $u \overset{*}{\Rightarrow} \widehat{u} \in \mathbb{F}$. This shows (70) and hence the lemma. □

Let us define a new formula Ψ_{Ben} in two steps:

(1) Every constraint $X \in \pi_E(L(\mathcal{B}))$ (resp. $X \notin \pi_E(L(\mathcal{B}))$ is replaced by $X \in L(\mathcal{B}')$ (resp. $X \notin L(\mathcal{B}')$ where $\mathcal{B}'$ is the NFA constructed in Lemma 14.12. Let Ψ be the new formula.
(2) Define Ψ_{Ben} by

$$\Psi_{\mathrm{Ben}} = \Psi \wedge \bigwedge \{X \in \mathbb{F} \mid X \in \mathcal{X}\}\,. \tag{71}$$

Lemma 14.13. *Let Φ satisfy the properties in Lemma 14.5. Then the embedding $\varphi\colon G \to \mathbb{F}(E) \rtimes H$ induces a bijection between $\pi_A(\mathrm{Sol}_{A,k}(\Phi)) \subseteq \mathbb{F}(A)^k$ and $\pi_E(\mathrm{Sol}_{E,k}(\Psi_{Ben})) \subseteq \mathbb{F}(E)^k$. Moreover, $(\|\Psi_{Ben}\|_{eq}, \|\Psi_{Ben}\|_{rat}) \in \mathcal{O}(\|\Phi\|_{eq}, \|\Phi\|_{rat})$.*

Proof. The proof is immediate by (69), (70), and the construction of Ψ_{Ben} which makes sure that all variables satisfy the constraint $X \in \mathbb{F}$. Thus a constraint $X \in L(\mathcal{B}')$ is equivalent to a constraint $X \in L(\mathcal{B}') \cap \mathbb{F}$ and a constraint $X \notin L(\mathcal{B}')$ is equivalent to a constraint $X \in \mathbb{F} \setminus L(\mathcal{B}')$. □

14.5. **Proof of Theorem 14.2, Phase 5.**

Switching from NFAs to finite monoids: From $\Psi_{\mathbf{Ben}}$ to $\Psi_{\mathbf{mon}}$. The goal is to reduce the proof Theorem 14.2 to Theorem 4.3. This requires that we represent regular constraints by recognizing morphisms. In the following a *guess* means to run deterministically over all possibilities. That is, there is deterministic transducer which respects the space bound in Theorem 14.2 and produces all possible outputs one after another. The corresponding EDT0L relations are calculated separately and then everything is put together as we did when we split $\widetilde{\Phi}$ into formulae $\Phi_{\eta,j}$ in Equation (63).

Let $L(\mathcal{B}_1), \ldots, L(\mathcal{B}_\ell)$ be the list of NFAs which appear in Ψ_{Ben}. We have $\ell \geqslant 1$ and without restriction $L(\mathcal{B}_\ell) = \mathbb{F}$. According to Example 2.1 there is a morphism $\mu_\ell \colon E^* \to N_\ell$ to an H-monoid N_ℓ of size $2+|E|^2-|E|$ such that $u \in \mathbb{F} \iff \mu_\ell(u) \in F_\ell$ where $F_\ell = \mu_\ell(\mathbb{F})$. Since $|E| \in \mathcal{O}(1)$, the monoid N_ℓ is of constant size. For the other constraints we cannot expect such a small recognizing monoid, and we use Boolean matrices instead. For $1 \leqslant i < \ell$ let q_i be the number of states of the NFA $\mathcal{B}_i$. According to Section 2.6 and Example 3.1 in Section 3 we find for each $1 \leqslant i < \ell$ a morphism to a morphism $\mu_i \colon E^* \to N_i$ to a monoid with involution N_i of size $4^{q_i^2}$ such that $u \in L(\mathcal{B}_i) \iff \mu_\ell(u) \in F_i$ where $F_i = \mu_i(\mathbb{F} \setminus L(\mathcal{B}_i))$ for a negative constraint and $F_i = \mu_i(L(\mathcal{B}_i))$ for a positive constraint. Recall that the monoid N_i is a submonoid of $\mathbb{B}^{2n \times 2n}$. Let N be the direct product $N_0 = N_1 \times \cdots \times N_\ell$. Let $\pi_i \colon N_0 \to N_i$ the canonical projection, Then we obtain a single morphism $\mu \colon E^* \to N_0$ such that $\mu_i = \pi_i \mu$ for all $1 \leqslant i < \ell$.

Now, for each $X \in \mathcal{X}$ we guess a value $\nu(X) \in N_0$. Each time we make a guess $\nu(X) \in N_0$ we check that it is consistent with the constraints. Thus, for each $0 \leqslant i \leqslant \ell$ and each $X \in \mathcal{X}$ we do the following. If there there a positive constraint $X \in \mathcal{B}_i$, then we check $\pi_i \nu(X) \in \mu_i(L(\mathcal{B}_i))$. If there is a negative constraint $X \notin \mathcal{B}_i$, then we check $\pi_i \nu(X) \notin \mu_i(L(\mathcal{B}_i))$. If the guess is not consistent, then the guess is not successful and the corresponding output is empty.

For a consistent guess ν we define the following formula

$$\Psi_{\mathrm{mon},\nu} = \bigwedge \{U_j = V_j \mid j \in J\} \wedge \bigwedge \{X = \nu(X) \mid X \in \mathcal{X}\}. \tag{72}$$

Here, $\{U_j = V_j \mid j \in J\}$ is the set of equations which appear in the conjunction Ψ_{Ben}. By a slight abuse of language we call a conjunction as in Equation (72) still a Boolean formula. It is clear what we mean by a *solution* of Ψ_{mon}, it is given by morphism $\sigma \colon \mathcal{X} \to E^*$ such that

(1) $\pi_E \sigma(U_j) = \pi_E \sigma(V_j) \in \mathbb{F}(E)$ for all $j \in J$.
(2) $\mu\sigma(X) = \nu(X)$ for all $X \in \mathcal{X}$.

For an inconsistent guess we let $\Psi_{\mathrm{mon},\nu} = \bot$. Using this interpretation we have

$$\mathrm{Sol}_{E,k}(\Psi_{\mathrm{Ben}}) = \bigcup \left\{ \mathrm{Sol}_{E,k}(\Psi_{\mathrm{mon},\nu}) \;\middle|\; \nu \in N_0^{\mathcal{X}} \right\}. \tag{73}$$

The size of the finite monoid N_0 is in $2^{\mathcal{O}(\|\Phi\|^2_{\mathrm{rat}})}$. Thus, in general we cannot store the disjunction over all guesses in PSPACE. So, we produce the required NFAs for each $\Psi_{\mathrm{mon},\nu}$ again one after another. We are approaching our goal prove Theorem 14.2. For that we use the following proposition.

Proposition 14.14. *Let Φ satisfy all conditions in Lemma 14.5. Then there is an* $\mathsf{NSPACE}(\|\Phi\|^2_{eq}(\|\Phi\|^2_{rat} + \log \|\Phi\|_{eq}))$ *algorithm which performs the following task. It takes as input a Boolean formula*

$$\Psi_{mon} = \bigwedge \{U_j = V_j \mid j \in J\} \wedge \bigwedge \{X = \nu(X) \mid X \in \mathcal{X}\} \tag{74}$$

which appears as $\Psi_{mon,\nu}$ in (72), The output is an extended alphabet C of size $\mathcal{O}(\|\Phi\|^2)$, letters $d_i \in C$ for all $1 \leqslant i \leqslant k$, and a trim NFA $\mathcal{A}_{\Psi_{mon}}$ accepting a rational set of S-morphisms over C^* such that the EDT0L relation

$$\{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{\Psi_{mon}})\}$$

is equal to the full solution set in freely reduced words

$$\mathrm{Sol}_{E,k}(\Psi_{mon}) = \left\{ (\sigma(X_1), \ldots, \sigma(X_k)) \in \mathbb{F}^k \;\middle|\; \pi_A \sigma(\Psi_{mon}) = \text{ true} \right\}.$$

Moreover, $\mathrm{Sol}_{E,k}(\Psi_{mon}) = \emptyset$ *if and only if* $L(\mathcal{A}_{\Psi_{mon}}) = \emptyset$*; and* $|\mathrm{Sol}_{E,k}(\Psi_{mon})| < \infty$ *if and only if* $\mathcal{A}_{\Psi_{mon}}$ *doesn't contain any directed cycle.*

14.6. **Proof of Proposition 14.14.** In the proof of Proposition 14.14 we wish to apply Theorem 4.3. In order to do so, we define another parameter $m(\Phi)$ by the following equation.

$$m(\Phi) \cdot \log(\|\Phi\|_{\text{eq}}) = \|\Phi\|_{\text{rat}}^2 . \tag{75}$$

Recall that $\log(m) \geqslant 1$ by the definition in Section 2.1. Therefore we have:

$$\mathsf{NSPACE}(\|\Phi\|_{\text{eq}}^2 (\|\Phi\|_{\text{rat}}^2 + \log\|\Phi\|_{\text{eq}})) = \mathsf{NSPACE}(\|\Phi\|_{\text{eq}}^2 \, m(\Phi) \log\|\Phi\|_{\text{eq}}). \tag{76}$$

Hence, for the proof we can use the space bound $\mathsf{NSPACE}(\|\Phi\|_{\text{eq}}^2 \, m(\Phi) \log\|\Phi\|_{\text{eq}})$ which is in the form we need for Theorem 4.3.

14.6.1. *From Ψ_{mon} to the system $\mathcal{S}_\Phi$.* Let Ψ_{mon} be written as in (74). Then we define $\mathcal{S}'$ to be the following system of equations (without any constraint)

$$\mathcal{S}' = \bigwedge \{U_j = V_j \mid j \in J\} .$$

Recall that we have defined morphisms $\mu\colon E^* \to N_0$ and $\nu : \mathcal{X} \to N_0$. We join μ and ν to a morphism $\mu_0\colon (E \cup \mathcal{X})^* \to N_0$ by letting $\mu_0(e) = \mu(e)$ for $e \in E$ and $\mu_0(X) = \nu(X)$ for $X \in \mathcal{X}$.

The group H acts on E, but neither on $\mathcal{X}$ nor on N_0. Therefore we perform two steps. First, we embed the set of variables $\mathcal{X}$ into a larger set of *twisted* variables

$$\mathcal{Y} = H \times \mathcal{X}.$$

In order to have an embedding we identify $Z \in \mathcal{X}$ with $(1, Z) \in \mathcal{Y}$.

The group H acts freely without fixed points on $\mathcal{Y}$ by $g \cdot (f, X) = (fg, X)$ and $\overline{(f, X)} = (f, \overline{X})$. In this way every morphism $\sigma\colon \mathcal{X} \to E^*$ extends uniquely to an H-compatible morphism $\sigma\colon \mathcal{Y}^* \to E^*$.

Second, we embed N_0 into a larger H-monoid N as constructed in Section 3.2. Moreover, using the universal property of the H-monoid N, we extend $\mu_0 : (E \cup \mathcal{X})^* \to N_0$ uniquely to a morphism $\mu_N : (E \cup \mathcal{Y})^* \to N$ of H-monoids by $\mu_N(f, Z) = f(\mu_0(Z))$. Twisted variable of the form (f, Z) appear in $\mathcal{S}'$ only if $f = 1$, but formally the set of variables is now $\mathcal{Y} = H \times \mathcal{X}$ and for each variable $Y \in \mathcal{Y}$ the value $\mu_N(Y) \in N$ is defined. The morphism μ_N is respects the involution and the action of H, so does every solution $\sigma\colon \mathcal{Y}^* \to E^*$.

We define the system $\mathcal{S}_\Phi$ by the system $\mathcal{S}'$ with the set of variables $\mathcal{Y}$ and where for each $Y \in \mathcal{Y}$ there is a constraint $\mu_N(Y) \in N$.

14.6.2. *Triangulation: From $\mathcal{S}_\Phi$ to $\mathcal{S}_{tri}$.* The "problem" with the system $\mathcal{S}_\Phi$ is that equations $U = V$ are written as words $U, V \in (\Gamma \cup \mathcal{Y})^*$ where $\Gamma = E \cup H \subseteq \mathbb{F}(E) \rtimes H$. In a twisted word equation the U and V should be words over $E \cup \mathcal{Y}$.

Now, let $W \in (\Gamma \cup \mathcal{Y})^*$ be any word. We intend to move letters $g \in H$ to the right. If W contains a factor fg with $f, g \in H$, then we replace fg by the letter h if $fg = h$ in H. Whenever we see a letter $h = 1 \in H$, then we remove it. If W contains a factor ga where $a \in E$ and $g \in H$, then we replace it by bg where $b \in E$ corresponds to the letter $ga\overline{g} \in E$ according to (65). Since b is letter g moves to the right without increasing the length of W. The last rule is that we replace every factor $g(f, Y)$ with $Y \in \mathcal{X}$ and $g \in H'$ by $(gf, Y)g$. Again, g moves to the right without increasing the length. Thanks to this rule twisted variables other than $(1, Z)$ appear in the equations. Thus, every $U = V$ with $U, V \in (\Gamma \cup \mathcal{Y})^*$ can be written as $U'f = V'g$ such that $U'V' \in (E \cup \mathcal{Y})^*$, $f, g \in H$, and $|U'V'| \leqslant |UV|$.

Moreover, if $\sigma\colon \mathcal{Y} \to E^*$ is any morphism, then $\sigma(U) = \sigma(U')f$ and $\sigma(U) = \sigma(U')g$. Since $\sigma(U') \in E^*$ and $\sigma(V') \in E^*$, we have $\pi_E\sigma(U) = \pi_E\sigma(V)$ only if $f = g$. Thus, whenever we find $f \neq g$, then we can stop: there is no solution. On the other hand, for $f = g$ we have $\pi_E\sigma(U) = \pi_E\sigma(V) \iff \pi_E\sigma(U') = \pi_E\sigma(V')$. Hence, we can replace the equation $U = V$ by $U' = V'$ and $U' = V'$ is a twisted word equation over E in twisted variables $\mathcal{Y} = H \times \mathcal{X}$ and with regular constraints defined by an compatible morphism $\mathcal{Y} \to N$.

Using standard techniques as described in Section 5.4 we can assume without restriction that all equations are in triangular form. The number of fresh (twisted) variables and the increase of the length over all equations is thereby bounded by $\mathcal{O}(\|\Phi\|_{\mathrm{eq}})$. The set of variables is still called $\mathcal{X}$ and the set twisted variables is still called $\mathcal{Y} = H \times \mathcal{X}$. Thus after the modifications above and triangulation we obtain the system of twisted word equation with regular constraints $\mathcal{S}_{\mathrm{tri}}$. We have $\varphi(\mathrm{Sol}_{A,k}(\Phi)) = \mathrm{Sol}_{E,k}(\mathcal{S}_{\mathrm{tri}})$ and all equations $\mathrm{Sol}_{E,k}(\mathcal{S}_{\mathrm{tri}})$ have the form

$$(1, Z) = f(x)g(y) \tag{77}$$

where $Z \in \mathcal{X}$ is a variable, $f, g \in H$, and $x, y \in E \cup \mathcal{Y}$. For example, we could have $x = e \in E$ and $y = (h, Y) \in \mathcal{Y}$. Then the equation becomes $(1, Z) = f(e)g((h, Y)) = e'(gh, Y)$. The triangular form is convenient to achieve the property that solutions are in freely reduced words.

14.6.3. *From $\mathcal{S}_{tri}$ to $\mathcal{S}_{fin}$: solutions in freely reduced words.* This subsection mimics [10] in the context of virtually free groups. $\mathcal{S}_{\mathrm{fin}}$ will be the "final" system in the sequence of transformations. Recall that $\mathbb{F} \subseteq E^*$ denotes the regular subset of freely reduced words. Clearly, if $puq \in \mathbb{F}$ and $f \in H$, then $f(u) \in \mathbb{F}$, too. Another crucial observation is that for all freely reduced words $x, y, z \in \mathbb{F}$ and $f, g \in H$ we have $z = f(x)g(y)$ in $\mathbb{F}(E)$ if and only if there are freely reduced words p, q, r such that

$$z = pr \quad \wedge \quad x = f^{-1}(p)q \quad \wedge \quad y = (g^{-1}f)(\overline{q})g^{-1}(r).$$

According to (77) every equation in the triangular system $\mathcal{S}_{\mathrm{tri}}$ has the form $(1, Z) = f(x)g(y)$. For each such equation and each $h \in H$ we introduce six fresh twisted variables

$$(h, P), (h, \overline{P}), (h, Q), (h, \overline{Q}), (h, R), (h, \overline{R})$$

After that we replace the equation $(1, Z) = f(x)g(y)$ by the conjunction of three new equations:

$$(1, Z) = (1, P)(1, R) \quad \wedge \quad x = (f, P)(1, Q) \quad \wedge \quad y = (g^{-1}f, \overline{Q})\,(g^{-1}, R). \tag{78}$$

For simplicity, the new set of twisted variables is still called $\mathcal{Y} = H \times \mathcal{X}$.

We obtain a system $\mathcal{S}_{\mathrm{fin}}$, and this finishes the construction of the new formula $\mathcal{S}_{\mathrm{fin}}$. Let $\sigma \colon \mathcal{X} \to E^*$ be any compatible morphism such that $\pi_E\sigma(1, Z) = \pi_E\sigma(f(x)g(y))$. Then we there is some $\sigma' : \mathcal{X} \to \mathbb{F}$ such that first, $\pi_E\sigma = \pi_E\sigma'$, and second σ' solves the three equation in (78) in freely reduced words. That is σ' solves the three equation under the constraint $(h, Y) \in \mathbb{F}$ for all (h, Y). For the other direction: if $\pi_E\sigma$ solves the three equations in $\mathbb{F}(E)$ without any constraint on twisted variables, then there is some $\sigma' \colon \mathcal{X} \to \mathbb{F}$ such that $\pi_E\sigma'$ solves the equation $(1, Z) = f(x)g(y)$ in $\mathbb{F}(E)$. The remaining problem is that our formalism asks to define values μ_N for each new variables. (That is 2/3 of all variables). The only way to do so in the given space bound is to guess the correct value. We can write the equations appearing in $\mathcal{S}_{\mathrm{fin}}$ as a system

$$\bigwedge \{x_j = (f_j, P_j)(g_j, R_j) \mid j \in J\} \tag{79}$$

where $x_j \in E \cup \mathcal{X}$ and $\mu_N(x_j)$ is already fixed. We can read this system as a system of equations over the finite monoid N_M. To check whether such systems have a solution is actually PSPACE hard, however we don't need PSPACE-hardness. It is enough that within our space bound we can output by guessing-and-checking all possibilities to assign μ_N values to each of the fresh twisted variables. Such an assignment is again a tuple $\nu \in N_M^{\mathcal{X}}$. Formally we can write a $\mathcal{S}_{\text{fin}}$ as a disjunction

$$\mathcal{S}_{\text{fin}} = \bigvee \left\{ \mathcal{S}_{\text{fin},\nu} \;\middle|\; \nu \in N_M^{\mathcal{X}} \right\}. \tag{80}$$

Some of the systems $\mathcal{S}_{\text{fin},\nu}$ can be empty. If all are empty, then we can stop: $\mathcal{S}_{\text{fin}}$ is not solvable.

Lemma 14.15. *Let Φ satisfy all conditions in Lemma 14.5. Then there is an $\mathsf{NSPACE}(\|\Phi\|_{eq}^2(\|\Phi\|_{rat}^2 + \log\|\Phi\|_{eq}))$ algorithm which performs the following task. It takes as input a Boolean formula non-empty system $\mathcal{S}_{fin,\nu}$ from the disjunction in Equation (80).*

The output is an extended alphabet C of size $\mathcal{O}(\|\Phi\|_{eq}^2)$ with $E \subseteq C$, letters $d_i \in C$ for all $1 \leqslant i \leqslant k$, and a trim NFA $\mathcal{A}_{\mathcal{S}_{fin,\nu}}$ accepting a rational set of E-morphisms over C^* such that the EDT0L relation

$$\left\{ (h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \;\middle|\; h \in L(\mathcal{A}_{\mathcal{S}_{fin,\nu}}) \right\}$$

is equal to the full solution set in freely reduced words

$$\mathrm{Sol}_{E,k}(\mathcal{S}_{fin,\nu}) = \left\{ (\sigma(X_1), \ldots, \sigma(X_k)) \in \mathbb{F}^k \;\middle|\; \pi_E\sigma \text{ solves } \mathcal{S}_{fin,\nu} \right\}.$$

Moreover, $\mathrm{Sol}_{E,k}(\mathcal{S}_{fin,\nu}) = \emptyset$ if and only if $L(\mathcal{A}_{\mathcal{S}_{fin,\nu}}) = \emptyset$; and $|\mathrm{Sol}_{E,k}(\mathcal{S}_{fin,\nu})| < \infty$ if and only if $\mathcal{A}_{\mathcal{S}_{fin,\nu}}$ doesn't contain any directed cycle.

Proof. The existence of the NFA $\mathcal{A}_{\mathcal{S}_{\text{fin},\nu}}$ with the desired properties is a formal consequence of Theorem 4.3. For the complexity issues we need an estimation of $m_{\mathcal{A}_{\mathcal{S}_{\text{fin},\nu}}}(N_M)$. It is however clear from the construction that we have $m_{\mathcal{A}_{\mathcal{S}_{\text{fin},\nu}}}(N_M) \in \mathcal{O}(m(\Phi))$ where $m_{\mathcal{A}_{\mathcal{S}_{\text{fin},\nu}}}(N_M)$ was defined in Equation (4) and $m(\Phi)$ was defined in Equation (75).

$\mathsf{NSPACE}(\|\Phi\|_{\text{eq}}^2(\|\Phi\|_{\text{rat}}^2 + \log\|\Phi\|_{\text{eq}})) = \mathsf{NSPACE}(\|\Phi\|_{\text{eq}}^2 m(\Phi) \log\|\Phi\|_{\text{eq}})$ is due to Equation (76). Thus the complexity follows again by Theorem 4.3. □

We did various modifications to the input formula Φ to arrive at a system $\mathcal{S}_{\text{fin},\nu}$ mentioned Lemma 14.15. Each step on the way from Φ to $\mathcal{S}_{\text{fin},\nu}$ involved a splitting or guessing, which are realized by transducers respecting the space bound. In order to define the NFA $\mathcal{A}_{\Psi_{\text{mon}}}$ which is needed for Proposition 14.14, we put all the pieces together. Thus, Proposition 14.14 is shown.

Corollary 14.16. *Let G be a finitely generated virtually free group given by a short exact sequence as in (58) and let $\varphi\colon G \to \mathbb{F}(E) \rtimes H$ the embedding of G into a semi-direct product as in Figure 9.*

Then there is an $\mathsf{NSPACE}(\|\Phi\|_{eq}^2(\|\Phi\|_{rat}^2 + \log\|\Phi\|_{eq}))$ algorithm which performs the following task. It takes as input a Boolean formula Φ. The output is an extended alphabet C of size $\mathcal{O}(\|\Phi\|_{eq}^2)$ with $E \subseteq C$, letters $d_i \in C$ for all $1 \leqslant i \leqslant k$, and a trim NFA $\mathcal{A}_{E,\Phi}$ accepting a rational set of E-morphisms over C^*. The corresponding EDT0L relation

$$\mathcal{R}(\mathcal{A}_{E,\Phi}) = \{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_{E,\Phi})\}$$

satisfies the following properties.

1. *We have $\mathcal{R}(\mathcal{A}_{E,\Phi}) \subseteq \mathbb{F}^k$. Thus, each for each $h \in L(\mathcal{A}_{E,\Phi})$ and $1 \leqslant i \leqslant k$ the word $h(d_i)$ is freely reduced.*
2. *We have $\varphi(\mathrm{Sol}_{A,k}(\Phi)) = \mathcal{R}(\mathcal{A}_{E,\Phi})$.*

Proof. As above we can use the same techniques of splitting and guessing based on (69), (71), and (73). Hence it is possible to construct the NFA $\mathcal{A}_{E,\Phi}$ by putting exponentially many NFAs of the form $\mathcal{A}_{\Psi_{\text{mon}}}$ provided by Proposition 14.14. Again we may use a transducer which satisfies the required space bound since all pieces can be constructed one after another. □

$$\begin{tikzcd}
A^* \arrow[r, "\varphi"] \arrow[d, "\pi_A"] & T^*AT^* \arrow[r, "\psi"] \arrow[d, "\pi_E"] & A^* \arrow[d, "\pi_A"] \\
\mathbb{F}(A) \arrow[r, "\varphi"] & \mathbb{F}(E) \arrow[r, "\psi"] & \mathbb{F}(A)
\end{tikzcd}$$

FIGURE 11. $\psi\varphi = \mathrm{id}_A$ and ψ maps freely reduced words to freely reduced words by Corollary 14.9.

14.7. **Proof of Theorem 14.2. From the NFA $\mathcal{A}_{E,\Phi}$ back to Φ.** We have $A \subseteq E$ and in Section 14.2 we defined an A-morphism $\psi\colon E^* \to A^*$ by $\psi(t) = 1$ for all $t \in E \setminus A$. Since $\varphi(a) \in T^*aT^*$ we see $\psi\varphi(a) = a$ for all $a \in A$. See the commutative diagram in Figure 11. Therefore, the second statement in Corollary 14.16 yields

$$\mathrm{Sol}_{A,k}(\Phi) = \psi(\mathcal{R}(\mathcal{A}_{E,\Phi})).$$

The first statement says that $\mathcal{R}(\mathcal{A}_{E,\Phi})$ is an EDT0L relation in freely reduced words over E; and Corollary 14.9 asserts that ψ maps freely reduced words to freely reduced words over A. Using one state more than the NFA $\mathcal{A}_{E,\Phi}$ (actually a new initial state) and a transition labeled by ψ from the old initial state to the new one, we obtain the desired NFA $\mathcal{A}_{\Phi}$. Hence, we can realize $\mathrm{Sol}_{A,k}(\Phi)$ as an effective EDT0L relation in freely reduced words over A. Thus, the projection $\pi_A\colon A^* \to \mathbb{F}(A)$ yields a bijection between $\mathrm{Sol}_{A,k}(\Phi)$ and the full solution set $\pi_A(\mathrm{Sol}_{A,k}(\Phi)) \subseteq \mathbb{F}(A)^k$. This concludes the proof of Theorem 14.2.

15. $\mathrm{SL}(2,\mathbb{Z})$

We apply in this section our results to the perhaps most prominent example of a (non-free) virtually free group: the special linear group $\mathrm{SL}(2,\mathbb{Z})$ of 2×2 matrices over $\mathbb{Z}$. It is well known (and easy to show) that $\mathrm{SL}(2,\mathbb{Z})$ is the amalgamated product $\mathrm{SL}(2,\mathbb{Z}) = \mathbb{Z}/4\mathbb{Z} \star_{\mathbb{Z}/2\mathbb{Z}} \mathbb{Z}/6\mathbb{Z}$. Possible generators are the matrices $\rho = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 1 \end{smallmatrix}\right)$ and $\tau = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ of orders 6 and 4 respectively. We have $\rho^3 = \tau^2 = \left(\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. We also denote the matrices $\left(\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ as -1 and 1 respectively.

When working with equations over $\mathrm{SL}(2,\mathbb{Z})$ it is more natural that the constants are matrices (with entries written as binary numbers) rather than words over a finite generating set. Moreover, in an equation there is no reason to see a factor of two matrices $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)\left(\begin{smallmatrix} a' & b' \\ c' & d' \end{smallmatrix}\right)$ because we would multiply the matrices together. For a matrix $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ in $\mathrm{SL}(2,\mathbb{Z})$ we let $\|M\|_1 = \max\{|a| + |c|,\, |b| + |d|\}$; and we define its *binary size*

$$\|M\|_{\mathrm{bin}} = \log \|M\|_1 .$$

Note that $\|M\|_1$ is the usual matrix *one-norm* of the matrix $\left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$. We use the notion of binary size to define the size of equations and Boolean formulae where constants are matrices. The only difference is that the size of a constant M in $\mathrm{SL}(2,\mathbb{Z})$ is not 1 as for a finite generating set, but $\|M\|_{\mathrm{bin}}$. We leave it to the reader to define the size of a Boolean formula accordingly. To have a notation for Boolean formulae Φ as well, we denote the new size by $\|\Phi\|_{\mathrm{bin}}$.

The aim of this section is to prove the following result.

Corollary 15.1. *There exists a standard generating set S for $\mathrm{SL}(2,\mathbb{Z})$ of 21 letters, and an $\mathsf{NSPACE}(m(\Phi)\,\|\Phi\|_{bin}^2 \log \|\Phi\|_{bin})$ algorithm which performs the following task. It takes as input a Boolean formula Φ where the constants are matrices over*

$\mathrm{SL}(2,\mathbb{Z})$ *(counted in their binary size) and in variables from* $\mathcal{X} = \mathcal{X}_+ \cup \mathcal{X}_-$ *such that* $X \in \mathcal{X}_+ \iff \overline{X} \in \mathcal{X}_-$ *and* $\mathcal{X}_+ = \{X_1, \ldots, X_k\}$*, where each variable has size* 1 *for simplicity. The output is an extended alphabet* C *of size* $\mathcal{O}(\|\Phi\|_{bin}^2)$*, letters* $d_i \in C$ *for all* $1 \leqslant i \leqslant k$*, and a trim NFA* $\mathcal{A}_\Phi$ *accepting a rational set of A-morphisms over* C^* *such that the EDT0L relation*

$$\{(h(d_1), \ldots, h(d_k)) \in C^* \times \cdots \times C^* \mid h \in L(\mathcal{A}_\Phi)\}$$

is equal to the full solution set in standard normal form as given in Equation (57)

$$\mathrm{Sol}_{S,k}(\Phi) = \left\{(\sigma(X_1), \ldots, \sigma(X_k)) \in \mathrm{snf}_S(G)^k \;\middle|\; \pi\sigma(\Phi) = \mathit{true}\right\}.$$

Moreover, $\mathrm{Sol}(\Phi) = \emptyset$ *if and only if* $L(\mathcal{A}) = \emptyset$*; and* $|\mathrm{Sol}(\Phi)| < \infty$ *if and only if* $\mathcal{A}$ *doesn't contain any directed cycle.*

The proof of Corollary 15.1 covers the rest of the section. In a first part, we make the reduction to the framework of Theorem 14.2 fully explicit. The main message is that a few elementary facts about $\mathrm{SL}(2,\mathbb{Z})$ are enough to understand Theorem 14.2 without any reference to Bass-Serre theory [57] or the resulting black box Proposition 14.7. In fact, what we use about $\mathrm{SL}(2,\mathbb{Z})$ predates the invention of Bass-Serre theory by far. For that we reformulate Theorem 14.2 for $\mathrm{SL}(2,\mathbb{Z})$ in Corollary 15.1. The point is that we view Corollary 15.1 directly as a corollary to Theorem 4.3.

In the second part we show that working with matrices doesn't increase the complexity. To see the difference, let $w \in \left\{\rho, \rho^{-1}, \tau, \tau^{-1}\right\}^*$ be a word in the (symmetric) set of natural generators with $n = |w|_{\rho^{\pm 1}} = |w|_\rho + |w|_{\rho^{-1}}$, and let $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ denote its image in $\mathrm{SL}(2,\mathbb{Z})$, then a straightforward calculation shows $\left\|\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)\right\|_1 \leqslant F_{n+2}$, where F_{n+2} is the $(n+2)$nd Fibonacci number. In particular,

$$\left\|\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)\right\|_{\mathrm{bin}} \leqslant |w|_{\rho^{\pm 1}} \leqslant |w|.$$

This means that working with matrices and their binary size doesn't increase the input size with respect to the reduced word lengths over $\left\{\rho, \rho^{-1}, \tau, \tau^{-1}\right\}^*$. However, an exponential gap between $\left\|\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)\right\|_{\mathrm{bin}}$ and $|w|_{\rho^{\pm 1}}$ is possible. For example, we have $(\tau\rho)^n = \left(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right)$ and $\|\left(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right)\|_{\mathrm{bin}} = \log(n+1)$. It is easy to see that $(\tau\rho)^n$ is the shortest word in $\left\{\rho, \rho^{-1}, \tau, \tau^{-1}\right\}^*$ which represents the matrix $\left(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right)$. Thus, the matrix representation of (shortest) words can lead to an exponential compression. However, in [21] Gurevich and Schupp give an exponential representation of a matrix M in $\mathrm{SL}(2,\mathbb{Z})$ by words over $\{\rho, \tau\}^*$ where the bit complexity of the exponential representation is linear in $\|M\|_{\mathrm{bin}}$. (In an exponential representation exponents over factors are written in binary.) In order to prove the lemma of Gurevich and Schupp we use the matrices $L = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) = \tau\rho^2$ and $U = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) = \tau\rho$.

15.1. **Explicit embedding of $\mathrm{SL}(2,\mathbb{Z})$ into a semi-direct product.** There is a natural action of $\mathrm{SL}(2,\mathbb{Z})$ on the complete bipartite graph $K_{3,2}$ which is depicted in Figure 12. We give an orientation to the set of undirected edges in $K_{3,2}$ according to that picture[5]. Denoting the set of directed edges $\{a, \ldots, f\}$, we obtain an alphabet $E = \left\{a, \overline{a}, \ldots, f, \overline{f}\right\}$. As usual, an undirected edge is two-element set $\{y, \overline{y}\}$.)

The action of $\mathrm{SL}(2,\mathbb{Z})$ on the bipartite graph $K_{3,2}$ is as follows. The generator τ stabilizes the vertices P_α and we let $\tau R_\beta = R_{\tau\beta}$ (with $R_{\tau^2} = R_1$). The generator ρ stabilizes the vertices R_β and we let $\rho P_\alpha = P_{\rho\beta}$ (with $P_{\rho^3} = P_1$).

Let us define a surjective homomorphism $\gamma\colon \mathrm{SL}(2,\mathbb{Z}) \to \mathbb{Z}/12\mathbb{Z}$ by γ by $r = 2 = \gamma(\rho)$ and $t = 3 = \gamma(\tau)$. (It is known that the kernel of γ is the commutator subgroup

[5] Actually, $K_{3,2}$ is the quotient graph of the Bass-Serre tree for $\mathrm{SL}(2,\mathbb{Z})$ modulo the action by that group. Figure 12 appears in [13] as well as some subsequent calculations.

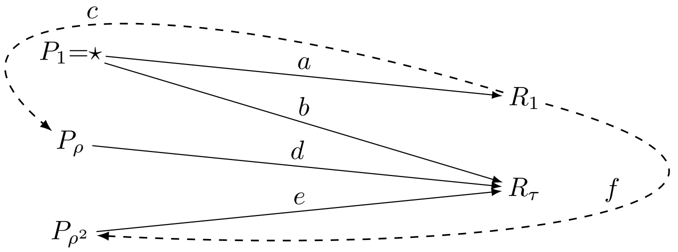

FIGURE 12. The complete bipartite graph $K_{3,2}$ with a directed spanning tree $\{a, b, d, e\}$ and set of directed chords c and f

of $\mathrm{SL}(2,\mathbb{Z})$; we'll come back to this in a minute.) An easy reflection shows that the action on on $K_{3,2}$ of $\mathrm{SL}(2,\mathbb{Z})$ factorizes through γ: indeed, $\mathbb{Z}/6\mathbb{Z} = \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ is quotient of the amalgamated product $\mathbb{Z}/3\mathbb{Z} \star_{\mathbb{Z}/2\mathbb{Z}} \mathbb{Z}/4\mathbb{Z}$. The action induces an action of $\mathbb{Z}/12\mathbb{Z}$ on the edge set E which respects the involution. For example: $\tau(c) = \overline{d}$, $\rho(\overline{c}) = d$, and $\rho(c) = f$ etc. Therefore, the homomorphism $H \to \mathrm{Aut}(E)$ yields a semi-direct product $\mathbb{F}(E) \rtimes H$. Note that the action of $H = \mathbb{Z}/12\mathbb{Z}$ on E is not faithful, the number 6 mod 12 ($= r^3 = t^2$) acts as the identity. This immediately implies that $\tau\rho\tau\rho^2$, and $\tau\rho^2\tau\rho$ are in the commutator subgroup $\mathrm{SL}(2,\mathbb{Z})'$.

A well-known classical result of Newman states that the kernel of $\gamma\colon \mathrm{SL}(2,\mathbb{Z}) \to \mathbb{Z}/12\mathbb{Z}$ is the commutator subgroup $\mathrm{SL}(2,\mathbb{Z})'$ and that $\mathrm{SL}(2,\mathbb{Z})'$ is a free group of rank 2 with basis $\left\{\tau\rho\tau\rho^2,\ \tau\rho^2\tau\rho\right\}$, [41, Lem. 1]. This is rather easy to see. A straightforward calculation shows that $[\tau,\rho] = \tau\rho\tau\rho^2$, $[\tau,\rho^2] = \tau\rho^2\tau\rho$ and that the subgroup generated by $\left\{\tau\rho\tau\rho^2,\ \tau\rho^2\tau\rho\right\}$ is closed under conjugation. Thus, $\mathrm{SL}(2,\mathbb{Z})'$ is generated by $\tau\rho\tau\rho^2$ and $\tau\rho^2\tau\rho$. In particular, let $F(c,f)$ denote the free group of rank 2 with basis $\{c,f\}$. Then $\psi(c) = \tau\rho\tau\rho^2$ and $\psi(f) = \tau\rho^2\tau\rho$ defines a surjective homomorphism

$$\psi\colon F(c,f) \to \mathrm{SL}(2,\mathbb{Z})'.$$

For the result of Newman it remains to show that $\mathrm{SL}(2,\mathbb{Z})'$ is freely generated by these two elements. We will do so by showing that $\mathrm{SL}(2,\mathbb{Z})'$ is the fundamental group of the graph $\Gamma = K_{3,2}$. Simultaneously, we will derive the desired result that $\mathrm{SL}(2,\mathbb{Z})$ embeds into the semi-direct product $\mathbb{F}(E) \rtimes H$ where $H = \mathbb{Z}/12\mathbb{Z}$.

We choose $\star = P_1$ as a base point in Γ. Then the fundamental group $\pi_1(\Gamma,\star)$ (which is, by definition, a subgroup in $\mathbb{F}(E) = F(a,b,c,d,e,f)$) can be identified with the free group $F(c,f)$. This identification is due to the fact that c, f are the chords for the chosen spanning (directed) tree $T = \{a,b,d,e\}$. Indeed, a standard exercise shows that the isomorphism $\varphi_1\colon F(c,f) \to \pi_1(\Gamma,\star)$ is given by

$$\varphi_1(c) = acd\overline{b} \text{ and } \varphi_1(f) = afe\overline{b}.$$

(To see this, say for c, just follow the shortest path in T from $\star$ to the source of c, traverse the chord c and choose the shortest path in T back to $\star$.) We have $\pi_1(\Gamma,\star) \leqslant \mathbb{F}(E)$ and a canonical projection $\tau\colon \mathbb{F}(E) \to F(c,f)$ which maps the edges of T to 1. Thus, $\tau\varphi_1$ is the identity on $F(c,f)$.

Finally, guided by $\tau(P_1) = P_1$ and $\rho(P_1) = P_\rho$ we define a homomorphism $\varphi\colon \mathrm{SL}(2,\mathbb{Z}) \to \mathbb{F}(E) \rtimes H$ where

$$\varphi(\tau) = (1,t) \text{ and } \varphi(\rho) = (b\overline{d}, r).$$

The homomorphism φ is well-defined since

$$\varphi(\tau^2) = (1,t^2) = (1,r^3) = \varphi(\rho^3) \text{ and } (1,t^2)^2 = (1,1).$$

$$\begin{tikzcd}
1 \arrow[r] & F(c,f) \arrow[r, "\psi"] \arrow[d, "\text{incl.}"] & \mathrm{SL}(2,\mathbb{Z}) \arrow[r] \arrow[d, "\varphi"] & \mathbb{Z}/12\mathbb{Z} \arrow[r] \arrow[d, "\mathrm{id}_{\mathbb{Z}/12\mathbb{Z}}"] & 1 \\
1 \arrow[r] & \mathbb{F}(E) \arrow[r] & \mathbb{F}(E) \rtimes \mathbb{Z}/12\mathbb{Z} \arrow[r] & \mathbb{Z}/12\mathbb{Z} \arrow[r] & 1
\end{tikzcd}$$

FIGURE 13. Embedding of $\mathrm{SL}(2,\mathbb{Z})$ into a semi-direct product.

Another direct calculation shows

$$\varphi\psi(c) = \varphi(\tau\rho\tau\rho^2) = (\varphi_1(c), 1) \text{ and } \varphi\psi(f) = \varphi(\tau\rho^2\tau\rho) = (\varphi_1(f), 1).$$

Thus, the identity $\mathrm{id}_{F(c,e)}$ factorizes as follows:

$$\mathrm{id}_{F(c,e)} : F(c,e) \xrightarrow{\psi} \mathrm{SL}(2,\mathbb{Z})' \xrightarrow{\varphi} \mathbb{F}(E) \times \{1\} = \mathbb{F}(E) \xrightarrow{\tau} F(c,e).$$

As a consequence, the surjection ψ is injective, and we obtain a commutative diagram Figure 13. The diagram implies that $\varphi\colon\ \mathrm{SL}(2,\mathbb{Z}) \to \mathbb{F}(E) \rtimes H$ is injective and that φ induces an isomorphism between $\mathrm{SL}(2,\mathbb{Z})'$ and $\pi_1(\Gamma, \star)$.

Since $\varphi(\mathrm{SL}(2,\mathbb{Z}))$ is a finitely generated subgroup, it is a rational subset in $\mathbb{F}(E) \rtimes H$. Hence, we can reduce the question about solving equations in $\mathrm{SL}(2,\mathbb{Z})$ to twisted word equations over $\mathbb{F}(E)$[6]. However, Theorem 4.3 is more ambitious than for example [7]. In order to apply Theorem 4.3 we need in particular an explicit construction of a set of standard generators. We obtain such a set S by defining $S = A_+ \cup A_- \cup H_+ \cup H_-$ where $A_+ = \{c, f\} = \left\{\tau\rho\tau\rho^2,\ \tau\rho^2\tau\rho\right\}$ and $H_+ = \left\{\rho^1, \dots, \rho^5,\ \tau, \rho^1\tau, \dots, \rho^5\tau\right\}$. We have $H_+ \cap H_- = \left\{\rho^1, \dots, \rho^5\right\}$ and ρ^3 becomes a self-involuting letter in S.

Remark 15.2. *Let $h = \rho^1\tau$, $h' = \rho^1$, and $g = \rho^2\tau$ be letters in H_+. Then the element $hh' \in S^*$ has length 2. The corresponding element in standard normal form is $\overline{c}fg \in A^*H_+$ which has length 3. This yields a concrete example showing that the standard normal forms are not geodesic in general.*

What remains to show for the proof of Corollary 15.1 is to show that the complexity is not worse than $\mathsf{NSPACE}(\|\Phi\|_{\text{bin}}^2\, m(\Phi) \log \|\Phi\|_{\text{bin}})$. This is done next.

15.2. **Euclidean matrix calculation.** We have $L^{-1} = \rho\tau = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 1 \end{smallmatrix}\right)$ and hence, $U^{-1}L = \rho$. Since ρ, τ generate $\mathrm{SL}(2,\mathbb{Z})$ as a monoid, we see that L, U generate $\mathrm{SL}(2,\mathbb{Z})$ as a group. It is therefore clear that every matrix in $\mathrm{SL}(2,\mathbb{Z})$ can be written as a word in $\left\{L, L^{-1}, U, U^{-1}\right\}^*$, but of course the representation is not unique as for example $(U^{-1}L)^6 = 1$.

Let $a_0, a_1 \in \mathbb{N}$ with $a_0 > a_1 > 0$. Using the extended Euclidean algorithm for computing the $\gcd(a_0, a_1)$ we define natural numbers k_i for $0 \leqslant i < g$ and a_i $0 \leqslant i \leqslant g+1$ with

$$a_0 > a_1 > \cdots > a_{g-1} > a_g = \gcd(a_0, a_1) > a_{g+1} = 0$$

such that for $i \geqslant 0$ we have

$$0 \leqslant a_{i+2} = a_i - k_i a_{i+1} < a_{i+1}. \tag{81}$$

The sequence finishes with some $1 \leqslant g \in \mathcal{O}(\log |a_0|)$ such that $k_{g-1}a_g = a_{g-1}$ and $a_g = \gcd(a_0, a_1)$. The last value is therefore indeed $a_{g+1} = 0$. We say that

[6] So far this is very close, if not identical, to the approach in [7].

$(k_0, \dots, k_{g-1})$ is the gcd-*sequence* defined by a_0, a_1. Note that $(k_0, \dots, k_{g-1})$ together with a_g uniquely define $(a_0, \dots, a_g)$. Note also that $k_i \geqslant 1$ for $0 \leqslant i < g-1$ and $k_{g-1} = a_{g-1} \geqslant 2$.

By $\mathrm{SL}(2,\mathbb{N})$ we mean the following submonoid of $\mathrm{SL}(2,\mathbb{Z})$:

$$\mathrm{SL}(2,\mathbb{N}) = \left\{ \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \;\middle|\; a,b,c,d \in \mathbb{N} \wedge ad - bc = 1 \right\}.$$

It is a well-known classical fact and not difficult to see that $\mathrm{SL}(2,\mathbb{N})$ is a free monoid with unique basis $\{U, L\}$, see for example [12, Chap. 8.12] or [28] for an application to fast randomized pattern matching. The following quantitative lemma belongs probably to folklore. It can be easily derived from [21], but for lack of a reference for the precise statement we give a proof.

Lemma 15.3. *Let* $M = \left(\begin{smallmatrix} a_0 & a_1 \\ c_0 & c_1 \end{smallmatrix}\right) \in \mathrm{SL}(2,\mathbb{N})$ *with* $a_0 > a_1 > 0$ *and let* $(k_0, \dots, k_{g-1})$ *be the* gcd-*sequence defined by* a_0, a_1. *Then there is a (unique)* $c_g \in \mathbb{N}$ *such that following assertions hold.*

(1) $0 < k_0 \cdots k_{g-1} \cdot \min\{1, c_g\} < a_0 + c_0 = \|M\|_1$.

(2) If g *is even, then*

$$M = L^{c_g} U^{k_{g-1}} L^{k_{g-2}} \cdots U^{k_1} L^{k_0}.$$

(3) If g *is odd, then* $c_g > 0$ *and*

$$M = L^{c_g - 1} U L^{k_{g-1}} \cdots U^{k_1} L^{k_0}.$$

Proof. For the following we don't need the uniqueness of c_g. It follows from the fact that $\{L, U\}$ forms a basis for the free monoid $\mathrm{SL}(2,\mathbb{N})$, which in turn follows easily from the present proof. We leave this part to the interested reader.

Consider a matrix $M_1 = M = \left(\begin{smallmatrix} a_0 & a_1 \\ c_0 & c_1 \end{smallmatrix}\right) \in \mathrm{SL}(2,\mathbb{N})$ with $a_0 > a_1 > 0$. Note that this implies $\gcd(a_0, a_1) = 1$. Moreover, $c_0 \geqslant c_1 > 0$ because $a_0 c_1 = a_1 c_0 + 1$. (The case $c_0 = c_1$ is possible only for $M = \left(\begin{smallmatrix} a_0 & a_0 - 1 \\ 1 & 1 \end{smallmatrix}\right)$.) Let us treat the case $a_1 = 1$ as a special case first. That is: $M = \left(\begin{smallmatrix} k_0 & 1 \\ c_0 & c_1 \end{smallmatrix}\right)$. We obtain $k_0 = a_0$ and

$$M = L^{c_1 - 1} U L^{a_0 - 1}.$$

Moreover, $c_0 = k_0 c_1 - 1$. Since $a_0 \geqslant 2$ we have $1 \leqslant c_0 < k_0 c_1 < a_0 + c_0$.

For the rest of the proof we may assume $g \geqslant 2$. We let $(k_0, \dots, k_{g-1})$ (and $(a_0, \dots, a_{g-1}, 1)$) be the gcd-sequences defined by a_0, a_1. Next, we define matrices M_i for $1 \leqslant i \leqslant g$ according to the following rules.

(1) If $1 \leqslant i < g$ and i is odd and $M_i = \left(\begin{smallmatrix} a_{i-1} & a_i \\ c_{i-1} & c_i \end{smallmatrix}\right)$ is defined, then we let

$$M_{i+1} = M_i L^{-k_{i-1}} = \left(\begin{smallmatrix} a_{i+1} & a_i \\ c_{i+1} & c_i \end{smallmatrix}\right).$$

(2) If $1 \leqslant i < g$ and i is even and $M_i = \left(\begin{smallmatrix} a_i & a_{i-1} \\ c_i & c_{i-1} \end{smallmatrix}\right)$ is defined, then we let

$$M_{i+1} = M_i U^{-k_{i-1}} = \left(\begin{smallmatrix} a_i & a_{i+1} \\ c_i & c_{i+1} \end{smallmatrix}\right).$$

It follows by induction that $M_i \in \mathrm{SL}(2,\mathbb{N})$ for all $1 \leqslant i \leqslant g$. Having this we can deduce, again by induction, for all $1 \leqslant i \leqslant g$:

$$0 < k_{i-1} a_i \leqslant a_{i-1} \tag{82}$$

$$0 < k_0 \cdots k_{i-1} a_i \leqslant a_0 \tag{83}$$

The situation for the c_i is slightly different. For for all $1 \leqslant i \leqslant g-1$

$$0 < k_{i-1} c_i \leqslant c_{i-1} \tag{84}$$

$$0 < k_0 \cdots k_{i-1} c_i \leqslant c_0 \tag{85}$$

$$0 \leqslant k_0 \cdots k_{g-2} \cdot \max\{c_{g-1},\, k_{g-1} c_g - 1\} \leqslant c_0 \tag{86}$$

To see (86) we observe that $1 \leqslant c_{g-1} = k_{g-1} c_g \pm 1$. Hence, we can use (85) to conclude (86). Considering $i = g-1$ shows the first claim in the lemma, because

(86) implies $k_0 \cdots k_{g-1} c_g \leqslant c_0 + k_0 \cdots k_{g-2}$ and $k_0 \cdots k_{g-2} < k_0 \cdots k_{g-1} \leqslant a_0$ by $k_{g-1} \geqslant 2$ and (83).

For the last matrix is M_g and depending on whether g is odd or even, we have two options. If g is even we let $D = L$ and $D = U$ otherwise. We obtain:

$$M_1 \cdot L^{-k_0} U^{-k_1} \cdots D^{-k_{g-2}} = M_g = \begin{cases} \left(\begin{smallmatrix} a_g & a_{g-1} \\ c_g & c_{g-1} \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & k_{g-1} \\ c_g & c_{g-1} \end{smallmatrix}\right) & \text{if } g \text{ is even;} \\ \left(\begin{smallmatrix} a_{g-1} & a_g \\ c_{g-1} & c_g \end{smallmatrix}\right) = \left(\begin{smallmatrix} k_{g-1} & 1 \\ c_{g-1} & c_g \end{smallmatrix}\right) & \text{if } g \text{ is odd.} \end{cases} \tag{87}$$

First case. Let g be even, hence $M_g = \left(\begin{smallmatrix} 1 & k_{g-1} \\ c_g & c_{g-1} \end{smallmatrix}\right)$. Then we have

$$M_g U^{-k_{g-1}} L^{-c_g} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right).$$

It is possible that $c_g = 0$ in the line above.
Second case. Let g be odd, hence $M_g = \left(\begin{smallmatrix} k_{g-1} & 1 \\ c_{g-1} & c_g \end{smallmatrix}\right)$. Then we have

$$M_g L^{1-k_{g-1}} U^{-1} L^{1-c_g} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right).$$

Note that for g odd, we have $c_g \geqslant c_g > 0$ and $k_{g-1} = a_{g-1} > a_g = 1$. Using (87) and a case distinction (whether or not g is even) yields the result. □

Proposition 15.4 (Gurevich and Schupp [21]). *Let $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}(2, \mathbb{Z})$ and $m = \max\{|a|, |b|, |c|, |d|\}$. Then there are words $u, v \in \{\rho, \tau\}^*$ and positive integers $e_0, \ldots, e_\ell$ with $0 \leqslant \ell \in \mathcal{O}(\log m)$ such that*

$$\begin{aligned} |uv| &\in \mathcal{O}(1), \\ 0 < e_0 \cdots e_\ell &< 2m \\ M &= u L^{e_0} U^{e_1} \cdots L^{e_{g-2}} U^{e_{\ell-1}} L^{e_\ell} v. \end{aligned}$$

Proof. As a preamble let us note that we will be able to enforce $e_0 \neq 0 \neq e_\ell$ because $L^{-1} = \rho\tau$ is a short word over ρ and τ.

The assertion is trivial for $m = 1$. Hence we assume $m \geqslant 2$. Using short words $u', v' \in \{\rho, \tau\}^*$, we obtain a matrix

$$M' = u' M v' = \left(\begin{smallmatrix} a_0 & a_1 \\ c_0 & c_1 \end{smallmatrix}\right)$$

with $m = a_0 > a_1 > 0$ and $c_0 > c_1 > 0$. Since $m \geqslant 2$ it is enough to see that we can choose $u' = \tau^{2+e_0} U^{e_1} \tau^{e_2}$ and $v' = \tau^{e_3} U^{e_4}$ where the exponents e_j are in $\{0, 1\}$. We have $M' \in \mathrm{SL}(2, \mathbb{N})$ and therefore the result follows from Lemma 15.3. □

Proof of Corollary 15.1. Proposition 15.4 shows that the size of the exponential expression

$$u L^{e_0} U^{e_1} \cdots L^{e_{g-2}} U^{e_{\ell-1}} L^{e_\ell} v$$

is linear in $\|M\|_{\mathrm{bin}}$. Thus, we can apply Corollary 14.16 based on the explicit embedding of $\mathrm{SL}(2, \mathbb{Z})$ into the semi-direct product as depicted in Figure 13.

Acknowledgments

Research supported by Australian Research Council (ARC) Project DP 160100486 and German Research Foundation (DFG) Project DI 435/7-1. The authors are indebted to the anonymous reviewer for their careful reading and extremely helpful feedback on this manuscript. We also thank Igor Potapov for pointing out the paper [21] of Gurevich and Schupp.

References

[1] P. R. Asveld. Controlled iteration grammars and full hyper-AFL's. *Information and Control*, 34(3):248 – 269, 1977.

[2] M. Benois. Parties rationelles du groupe libre. *C. R. Acad. Sci. Paris, Sér. A*, 269:1188–1190, 1969.

[3] V. Berthé, C. D. Felice, V. Delecroix, F. Dolce, J. Leroy, D. Perrin, C. Reutenauer, and G. Rindone. Specular sets. *Theoretical Computer Science*, 684:3–28, 2017.

[4] R. V. Book, T. J. Long, and A. L. Selman. Quantitative relativizations of complexity classes. *SIAM J. Comput.*, 13:461–487, 1984.

[5] L. Ciobanu, V. Diekert, and M. Elder. Solution sets for equations over free groups are EDT0L languages. *International Journal of Algebra and Computation*, 26:843–886, 2016. Conference abstract in ICALP 2015, LNCS 9135.

[6] L. Ciobanu and M. Elder. Solutions sets to systems of equations in hyperbolic groups are EDT0L in PSPACE. In C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 110:1–110:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[7] F. Dahmani and V. Guirardel. Foliations for solving equations in groups: free, virtually free and hyperbolic groups. *J. of Topology*, 3:343–404, 2010.

[8] V. Diekert. Makanin's Algorithm. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics and Its Applications*, chapter 12, pages 387–442. Cambridge University Press, 2002.

[9] V. Diekert and M. Elder. Solutions of twisted word equations, EDT0L languages, and context-free groups. In I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 96:1–96:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[10] V. Diekert, C. Gutiérrez, and Ch. Hagenah. The existential theory of equations with rational constraints in free groups is PSPACE-complete. *Information and Computation*, 202:105–140, 2005. Conference version in STACS 2001, LNCS 2010, 170–182, 2001.

[11] V. Diekert, A. Jeż, and W. Plandowski. Finding all solutions of equations in free groups and monoids with involution. *Information and Computation*, 251:263–286, 2016. Conference version in Proc. CSR 2014, LNCS 8476 (2014).

[12] V. Diekert, M. Kufleitner, G. Rosenberger, and U. Hertrampf. *Discrete Algebraic Methods. Arithmetic, Cryptography, Automata and Groups.* Walter de Gruyter, 2016.

[13] V. Diekert and A. Weiß. Context-Free Groups and Bass-Serre Theory. In J. González-Meneses, M. Lustig, and E. Ventura, editors, *Algorithmic and Geometric Topics Around Free Groups and Automorphisms*, Advanced Courses in Mathematics - CRM Barcelona. Birkhäuser, Basel, Switzerland, 2017.

[14] M. J. Dunwoody. The accessibility of finitely presented groups. *Inventiones Mathematicae*, 81(3):449–457, 1985.

[15] A. Ehrenfeucht and G. Rozenberg. On some context free languages that are not deterministic ET0L languages. *RAIRO Theor. Inform. Appl.*, 11:273–291, 1977.

[16] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, New York and London, 1974.

[17] J. Ferté, N. Marin, and G. Sénizergues. Word-mappings of level 2. *Theory Comput. Syst.*, 54:111–148, 2014.

[18] R. H. Gilman. Personal communication, 2012.

[19] R. H. Gilman, S. Hermiller, D. F. Holt, and S. Rees. A characterisation of virtually free groups. *Arch. Math. (Basel)*, 89(4):289–295, 2007.

[20] M. Gromov. Hyperbolic groups. In S. M. Gersten, editor, *Essays in Group Theory*, number 8 in MSRI Publ., pages 75–263. Springer-Verlag, 1987.

[21] Y. Gurevich and P. Schupp. Membership problem for the modular group. *SIAM J. Comput.*, 37:425–459, 2007.

[22] C. Gutiérrez. Satisfiability of equations in free groups is in PSPACE. In *Proceedings 32nd Annual ACM Symposium on Theory of Computing, STOC'2000*, pages 21–27. ACM Press, 2000.

[23] M. A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley Publishing Company, 1978.

[24] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[25] S. Jain, A. Miasnikov, and F. Stephan. The complexity of verbal languages over groups. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 405–414. IEEE Computer Society, 2012.

[26] A. Jeż. Recompression: a simple and powerful technique for word equations. *J. ACM*, 63(1):4:1–4:51, 2016.

[27] A. Jeż. Word equations in nondeterministic linear space. In I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, editors, *Proc. 44th International Colloquium Automata, Languages and Programming (ICALP 2017), Part II, Warsaw, Poland, July 10-14, 2017*, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[28] R. M. Karp and M. O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31:249–260, 1987.

[29] A. Karrass, A. Pietrowski, and D. Solitar. Finite and infinite cyclic extensions of free groups. *Journal of the Australian Mathematical Society*, 16(04):458–466, 1973.

[30] O. Kharlampovich and A. Myasnikov. Elementary theory of free non-abelian groups. *J. of Algebra*, 302:451–552, 2006.

[31] D. Kozen. Lower bounds for natural proof systems. In *Proc. of the 18th Ann. Symp. on Foundations of Computer Science, FOCS'77*, pages 254–266, Providence, Rhode Island, 1977. IEEE Computer Society Press.

[32] D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the group case. *Ann. Pure Appl. Logic*, 131(1-3):263–286, 2005.

[33] M. Lohrey. Algorithmics on SLP-compressed strings: A survey. *Groups Complexity Cryptology*, 4:241–299, 2012.

[34] M. Lohrey. *The Compressed Word Problem for Groups.* Springer Briefs in Mathematics. Springer, 2014.

[35] M. Lohrey and G. Sénizergues. Theories of HNN-extensions and amalgamated products. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP*, volume 4052 of *Lecture Notes in Computer Science*, pages 504–515. Springer, 2006.

[36] G. 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).

[37] G. S. Makanin. Decidability of the universal and positive theories of a free group. *Izv. Akad. Nauk SSSR*, Ser. Mat. 48:735–749, 1984. In Russian; English translation in: *Math. USSR Izvestija, 25*, 75–88, 1985.

[38] V. D. Mazurov and E. I. Khukhro, editors. *Unsolved Problems in Group Theory. The Kourovka Notebook. No. 10 (1986 edtion).* ArXiv 1401.0300, 2014.

[39] J. D. McKnight. Kleene quotient theorem. *Pacific Journal of Mathematics*, pages 1343–1352, 1964.

[40] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983.

[41] M. Newman. The structure of some subgroups of the modular group. *Illinois J. Math.*, 6:480–487, 1962.

[42] Ch. H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[43] W. Plandowski. Satisfiability of word equations with constants is in PSPACE. *J. ACM*, 51:483–496, 2004. Conference version in Proc. FOCS'99.

[44] W. Plandowski and W. 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.

[45] A. A. Razborov. *On Systems of Equations in Free Groups.* PhD thesis, Steklov Institute of Mathematics, 1987. In Russian.

[46] A. A. Razborov. On systems of equations in free groups. In *Combinatorial and Geometric Group Theory*, pages 269–283. Cambridge University Press, 1994.

[47] F. Rimlinger. Pregroups and Bass-Serre theory. *Mem. Amer. Math. Soc.*, 65(361):viii+73, 1987.

[48] E. Rips and Z. Sela. Canonical representatives and equations in hyperbolic groups. *Inventiones Mathematicae*, 120:489–512, 1995.

[49] G. Rozenberg and A. Salomaa. *The Book of L.* Springer, 1986.

[50] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages*, volume 1. Springer, 1997.

[51] Z. Sela. Diophantine geometry over groups VIII: Stability. *Ann. of Math.*, 177:787–868, 2013.

[52] A. L. Selman. A taxonomy of complexity classes of functions. *J. Comput. Syst. Sci.*, 48(2):357–381, 1994.

[53] A. L. Selman. Much ado about functions. In *Proc. 11th Annual IEEE Conference on Computational Complexity*, CCC '96, pages 198–212, Washington, DC, USA, 1996. IEEE Computer Society.

[54] G. Sénizergues. On the finite subgroups of a context-free group. In G. Baumslag, D. Epstein, R. Gilman, H. Short, and C. Sims, editors, *Geometric and Computational Perspectives on Infinite Groups*, number 25 in DIMACS series in Discrete Mathematics and Theoretical Computer Science, pages 201–212. Amer. Math. Soc., 1996.

[55] G. Sénizergues. On the rational subsets of the free group. *Acta Informatica*, 33:281–296, 1996.

[56] G. Sénizergues and A. Weiß. The isomorphism problem for finite extensions of free groups is in PSPACE. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 139:1–139:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[57] J.-P. Serre. *Trees.* Springer, 1980. French original 1977.

[58] P. V. Silva. Recognizable subsets of a group: finite extensions and the abelian case. *Bulletin of the EATCS*, 77:195–215, 2002.

Institut für Formale Methoden der Informatik, Universität Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, Germany
E-mail address: diekert@fmi.uni-stuttgart.de

School of Mathematical and Physical Sciences, University of Technology Sydney, Broadway NSW 2007, Australia
E-mail address: murray.elder@uts.edu.au