# Makanin's Algorithm for Solving Word Equations with Regular Constraints

**Article** · March 1998

DOI: 10.18419/opus-2419 · Source: OAI

Volker Diekert
Universität Stuttgart

Universität Stuttgart
Fakultät Informatik

Institut für Informatik
Breitwiesenstraße 20-22
D-70565 Stuttgart

# Makanin's Algorithm for Solving Word Equations with Regular Constraints

Volker Diekert

Report Nr. 1998/02

March 3, 1998

## Abstract

We give a self-contained proof of a fundamental result of Makanin (1977), which solves the satisfiability problem of equations with constants over free monoids. Our presentation of Makanin's algorithm is borrows Schulz (1992a), where Makanin's result is extended to the case where solutions are restricted by imposing regular constraints on the variables.

This report appears (with minor modifications) as a chapter of the new book of M. Lothaire *Algebraic Combinatorics on Words.*

# Contents

# 1 Introduction

The aim of this text is to give a self-contained proof of a fundamental result of Makanin which implies that the existential theory of word equations over free monoids is decidable. In other terms, Makanin's result is the decidability of the existential theory of concatenation. Let $A$ be an alphabet of constants and let $\Omega$ be a set of variables. A word equation $L = R$ is a pair $(L, R) \in (A \cup \Omega)^* \times (A \cup \Omega)^*$, and a *system of word equations* is a set of equations $\{L_1 = R_1, \ldots, L_k = R_k\}$. A *solution* is a homomorphism $\sigma : (A \cup \Omega)^* \to A^*$ leaving the letters of $A$ invariant such that $\sigma(L_i) = \sigma(R_i)$ for all $1 \leq i \leq k$. A solution is henceforth identified with a mapping $\sigma : \Omega \to A^*$. It is called *non-singular*, if $\sigma(x) \neq 1$ for all $x \in \Omega$. Otherwise it is called *singular*. The satisfiability problem for word equations is to decide whether a given word equation has a solution. The problem is usually stated for a single equation, but this is no loss of generality. Given a propositional formula over word equations all negations can be eliminated; and then, passing to a disjunctive normal form, the problem of satisfiability can be reduced to a single conjunction. It is therefore enough to consider a system of word equations, which in turn can be transformed into a single word equation. This way the decidability of propositional formulas over word equations can be reduced to the satisfiability problem of single word equations. Makanin (1977) yields a decision procedure for the satisfiability problem of word equations; and the decidability of the existential theory over free monoids follows. Since the problem is semi-decidable by its nature, a positive answer suffices to compute (if desired) a solution $\sigma : \Omega \to A^*$ which is minimal, say with respect to $\sum_{x\in\Omega} |\sigma(x)|$. Here and in the following *computation* means that there is an effective procedure in the mathematical sense. We shall derive a double exponential space bound only to solve the satisfiability problem, and the length of a minimal solution will be at most four times exponential in the input size of the word equation. These upper bounds are far beyond any practical meaning, but it is not clear that this reflects the inherent complexity of the problem. In practice the algorithm seems to behave much better. For example, up to now no solvable word equation is known where the minimal solution exceeds exponential length.

**Example 1.1** *Let $A = \{a, b\}$ and $\Omega = \{x, y, z, u\}$. Consider the equation*

$$xauzau = yzbxaaby$$

*This equation is solvable, a possible non-singular solution is given by:*

$$\sigma(x) = abb, \quad \sigma(y) = ab, \quad \sigma(z) = ba, \quad \sigma(u) = bab.$$

*We have*

$$abbababbaabab = \sigma(xauzau) = \sigma(yzbxaaby).$$

There is a rather straightforward algorithm to decide the solvability of a system of word equations where each variable occurs at most twice. This algorithm is due to Matiyasevich (1968). Since the general solution refers (implicitly) to the underlying idea, we explain it here as an introductory example. Let $E = \{L_1 = R_1, \ldots, L_k = R_k\}$ be a system of word equations where every variable $x \in \Omega$ occurs at most twice in $E$. Let $\|E\| = \sum_{i=1}^{k} |L_i R_i|$ denote the denotational length of $E$. The question is whether there is a solution. Using induction on $|\Omega|$ we describe a non-deterministic decision algorithm which works without exceeding a linear space bound in $\|E\|$. The basis $\Omega = \emptyset$ is clear, hence let $\Omega \neq \emptyset$. The first step is then to guess whether there is a solution $\sigma : \Omega \to A^*$ where $\sigma(x) = 1$ for some $x \in \Omega$. This is done by choosing some $x \in \Omega$ and replacing the occurrences of $x$ in $E$ by the empty word. We obtain a new system $E'$ over $\Omega \setminus \{x\}$ and recursively we decide in non-deterministic linear space whether $E'$ has a solution. Thus, after this step we are looking for non-singular solutions of $E$, only. We may assume that the first equation is either of the form

$$\begin{array}{llcll} & x\cdots & = & a\cdots & \text{with } x \in \Omega,\ a \in A \\ \text{or} & x\cdots & = & y\cdots & \text{with } x \in \Omega,\ y \in \Omega,\ x \neq y. \end{array}$$

By symmetry (or a non-deterministic guess to interchange the rôle of $L_1$ and $R_1$) we may either write $x = az$ or $x = yz$, where $z$ is a new variable. Replacing all occurrences of $x$ by $az$ or $yz$ respectively, we obtain a new system where $x$ does not occur any more and $z$ occurs at most twice. On the left of the first equation we may cancel either $a$ or $y$, and then $y$ occurs also at most twice. Hence we end up with a new system $E'$ where the number of variables is the same as in $E$, every variable occurs at most twice and we have $\|E'\| \leq \|E\|$. Note that $E'$ may have a singular solution with $\sigma(z) = 1$. However, if $E'$ is solvable, then $E$ is also solvable.

Now, let $\sigma : \Omega \to A^*$ be a non-singular solution of $E$ where $\sum_{x \in \Omega} |\sigma(x)|$ is minimal. Then we find a solution $\sigma'$ for $E'$ with $|\sigma'(z)| < |\sigma(x)|$ since $\sigma(y) \neq 1$. Thus, the length of a shortest solution has decreased, showing that the non-deterministic procedure will find a solution, if there is any. Since we have a linear space bound, the procedure can be transformed into a deterministic decision algorithm of at most exponential time.

The presentation of the general case will mainly follow Schulz (1992a), thereby showing the result of Makanin in a more general setting. Assume that for every $x \in \Omega$ a regular language $L_x \subseteq A^*$ is given as part of the problem instance together with the equation $L = R$. Then we can decide whether or not there exists a solution $\sigma : \Omega \to A^*$ satisfying additionally the regular constraints $\sigma(x) \in L_x$ for all $x \in \Omega$. (For example, we can prescribe the alphabet in a solution $\sigma(x)$ for all $x \in \Omega$.)

In the following we do not focus very much on necessary decidable conditions which are useful to prune the search tree. A good pruning strategy is of course extremely important for an implementation since the search tree tends to be huge. However pruning the tree doesn't help to understand the algorithm nor it seems to have any effect on the worst-case analysis.

# 2 Simple facts on words and on equations

## 2.1 Notations and combinatorial properties

Throughout this chapter $A = \{a, b, \dots\}$ denotes an alphabet of constants and $\Omega$ is a set of *variables* (or *unknowns*) such that $A \cap \Omega = \emptyset$. We shall use the same symbol $\sigma$ to denote a mapping $\sigma : \Omega \to A^*$ and its canonical extension to a homomorphism $\sigma : (A \cup \Omega)^* \to A^*$. The symbol 1 denotes the empty word and the unit element in other monoids and also the natural number $1 \in \mathbb{N}$. The length of a word $w$ is denoted by $|w|$. The prefix relation (proper prefix relation) of words is denoted by $u \leq v$ ( $u < v$ resp.). Recall that a word $p$ is called *primitive*, if it cannot be written in the form $p = r^\alpha$ with $\alpha \neq 1$. Lower case Greek letters $\alpha, \beta$ etc. are mostly used to denote natural numbers. The set of natural numbers is $\mathbb{N}$, it includes zero, the set of integers is $\mathbb{Z}$. By $\log \alpha$ we mean $\max\{1, \lceil \log_2 \alpha \rceil\}$.

Two words $y, z \in A^*$ are *conjugates*, if $xy = zx$ for some $x \in A^*$. The next proposition shows that in free monoids conjugates are obtained by transposition.

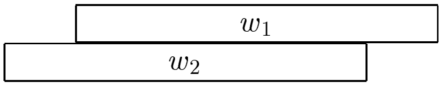
**Proposition 2.1** *Let $x, y, z \in A^*$ be words, $y, z \neq 1$. Then the following assertions are equivalent:*

1. $xy = zx$,
2. $\exists\, r, s \in A^*, s \neq 1, \alpha \geq 0 : x = (rs)^\alpha r,\ y = sr,$ *and* $z = rs$.

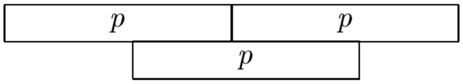
**Proposition 2.2** *Let $p \in A^*$ be primitive and $p^2 = xpy$ for some $x, y \in A^*$. Then we have either $x = 1$ or $y = 1$ (but not both).*

Proofs of Props. 2.1 and 2.2 can be found in Lothaire (1983) or elsewhere.

An overlapping of two words $w_1$ and $w_2$ is depicted by the following figure:

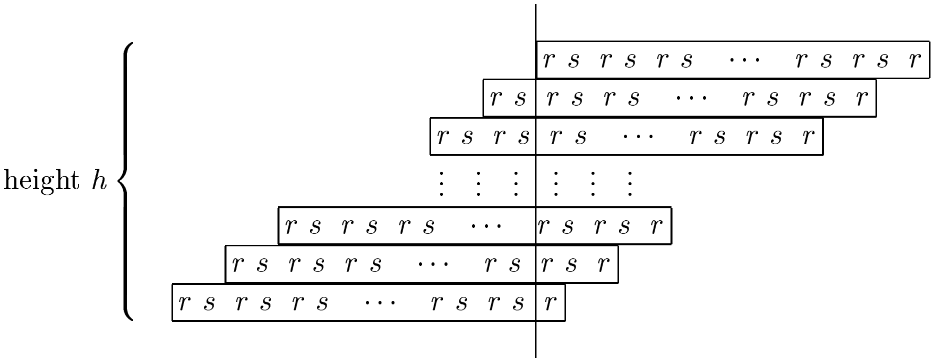


It says that the common border is an identical factor, i.e., $w_1 = xy$, $w_2 = zx$. Usually we mean $x \neq 1$ and sometimes the figure also indicates that both $y \neq 1$ and $z \neq 1$. But there will be no risk of confusion. For example, Prop. 2.2 can be rephrased by saying that the following picture is not possible for a primitive word $p \in A^*$:



## 2.2 Domino Towers

Every non-empty word $w \in A^+$ can be written in the form $w = (rs)^{h-1}r$ with $s \neq 1$, $h \geq 2$. Then it can be arranged in figure looking like a domino tower of height $h$:

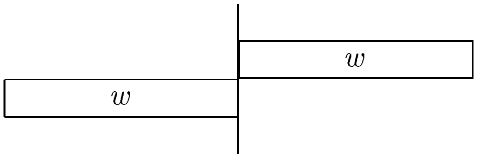

height h
r s r s r s ··· r s r s r
r s | r s r s ··· r s r s r
r s r s | r s ··· r s r s r
r s r s r s ··· | r s r s r
r s r s r s ··· r s | r s r
r s r s r s ··· r s r s | r

The position of the vertical line says that the upper left boundary is not on the right of the lower right boundary. The formal definition of such an arrangement allows also a less regular shape. It is as follows:

**Definition 2.3** *Let $h \geq 2$. We say that a non-empty word $w \in A^+$ can be arranged in a* domino tower of height *$h$, if there are words $x_1, \ldots, x_{h-1} \in A^*$ and non-empty words $y_1, \ldots, y_{h-1}$, $z_2, \ldots, z_h \in A^+$ such that*

1. *$w = x_i y_i = z_{i+1} x_i$ for all $1 \leq i < h$,*
2. *$|z_2 \cdots z_h| \leq |w|$.*

Note that for $h = 2$ the domino tower may degenerate as in the following figure.



**Definition 2.4** *Let $w \in A^*$ be a word. The* exponent of periodicity $\exp(w)$ *is defined by*

$$\exp(w) = \max\{\alpha \in \mathbb{N} \mid \exists r, s, p \in A^*, p \neq 1 : w = rp^\alpha s\}.$$

**Lemma 2.5** *Let $w \in A^+$ be a non-empty word which can be arranged in a domino tower of height $h$. Then we have $exp(w) \geq h \Leftrightarrow 1$.*

*Proof*. Choose a domino tower and words $x_i, y_i, z_i$ as in the definition above. Let $z = z_i \in \{z_2, \ldots, z_h\}$ be of minimal length, $x = x_{i-1}$, $y = y_{i-1}$. Then $(h \Leftrightarrow 1)|z| \le |w|$ and we have $xy = zx = w$. Hence $y$ and $z$ are conjugates and we may apply Prop. 2.1. We obtain $z = rs$ and $x = (rs)^\alpha r$ for some $\alpha \ge 0$ and $|r| < |z|$. Hence $w = z^{\alpha+1} r$ and therefore

$$(h \Leftrightarrow 1)|z| \le |w| < (\alpha + 2)|z| .$$

Since $|z| > 0$ we see that $h \Leftrightarrow 1 \le \alpha + 1 \le \exp(w)$. □

## 2.3 Stable normal forms

One of the key ideas in Makanin's proof is that, given a word equation, the exponent of periodicity of a shortest solution has an effective upper bound. This relies on the notion of *p-stable normal form.*

**Definition 2.6** *Let $p \in A^+$ be a primitive word. The $p$-*stable normal form *of the word $w \in A^*$ is a shortest sequence*

$$(u_0, \alpha_1, u_1, \ldots, \alpha_k, u_k)$$

*such that $k \ge 0$ ($k$ is minimal), $u_0, u_i \in A^*$, $\alpha_i \ge 0$ for $1 \le i \le k$, and the following conditions are satisfied:*

1. *We have $w = u_0 p^{\alpha_1} u_1 \cdots p^{\alpha_k} u_k$.*
2. *We have $k = 0$ if and only if $p^2$ is not a factor of $w$.*
3. *If $k \ge 1$, then we have:*

$$\begin{aligned} u_0 &\in A^*p \setminus A^*p^2A^*, \\ u_i &\in (A^*p \cap pA^*) \setminus A^*p^2A^* \text{ for } 1 \le i < k, \\ u_k &\in pA^* \setminus A^*p^2A^*. \end{aligned}$$

**Proposition 2.7** *Let $p \in A^+$ be primitive. The $p$-stable normal form of $w \in A^*$ is uniquely defined. This means, if $(u_0, \alpha_1, u_1, \ldots, \alpha_k, u_k)$ and $(v_0, \beta_1, \ldots, \beta_\ell, v_\ell)$ are $p$-stable normal forms of the same word $w \in A^*$, then they are identical, i.e., we have $k = \ell$, $u_0 = v_0$, $u_i = v_i$, and $\alpha_i = \beta_i$ for $1 \le i \le k$.*

*Proof*. Assume that $(u_0, \alpha_1, u_1, \ldots, \alpha_k, u_k)$ and $(v_0, \beta_1, v_1, \ldots, \beta_\ell, v_\ell)$ are both $p$-stable normal forms of $w$. Since these are shortest sequences, the indices $k$ and $\ell$ are both minimal, hence $k = \ell$.

For $k = 0$ we have $w = u_0 = v_0$, hence let $k = \ell \ge 1$.

We show first that $u_0 = v_0$. To see this, suppose by symmetry that $|u_0| \le |v_0|$. Since $u_0 p \in A^*p^2$ and $v_0 \in (A^*p \setminus A^*p^2A^*)$, we obtain that $u_0 \le v_0 < u_0 p$. By Prop. 2.2 this yields $u_0 = v_0$.

Let $w'$ denote the word $u_1 p^{\alpha_2} u_2 \cdots p^{\alpha_k} u_k$. A simple reflection using $u_1 \ne p$, Prop. 2.2, and $u_1 \in (A^*p \cap pA^*) \setminus A^*p^2A^*$ shows that $p^{\alpha_1} w' \in p^{\alpha_1+1}A^* \setminus p^{\alpha_1+2}A^*$. This implies $\alpha_1 = \beta_1$ and $w' = v_1 p^{\beta_2} v_2 \cdots p^{\beta_k} v_k$. Since we have $w' \in pA^*$, we see that the first component of its $p$-stable normal form is in $pA^*$. Hence $(u_1, \alpha_2, u_2, \ldots, \alpha_k, u_k)$ is the $p$-stable normal form of $w'$. By induction we conclude $(u_1, \alpha_2, u_2, \ldots, \alpha_k, u_k) = (v_1, \beta_2, v_2, \ldots, \beta_k, v_k)$. Hence the proposition. □

### 2.4 From a system to a single equation and vice versa

The existential theory of equations over free monoids is decidable, i.e., the satisfiability of any propositional formula over word equations (with regular constraints) can be decided. Let us show the reduction to Makanin's result. In a first step we may assume that all negations in a given formula are of type $L \neq R$. Due to the following proposition these negations can be eliminated.

**Proposition 2.8** *An inequality $L \neq R$ is equivalent with respect to satisfiability to the following formula using $x, y$ and $z$ as new variables:*

$$\bigvee_{a \in A} (L = Rax \vee R = Lax) \vee \bigvee_{\substack{a,b \in A \\ a \neq b}} (L = xay \wedge R = xbz)\,.$$

In a second step the formula (without negations) is written in disjunctive normal form. Then, for satisfiability, it is enough to see how a system of word equations can be transformed into a single word equation. The method is given in Prop. 2.9. It relies on the trivial fact that if $ua \leq va, ub \leq vb$, $u, v \in A^*$, $a, b \in A$, and $a \neq b$, then we have $u = v$.

**Proposition 2.9** *Let $a, b \in A$ be distinct letters (if $A = \{a\}$, then let $b$ denote a new letter resp.) and let $E = \{L_1 = R_1, \ldots, L_k = R_k\}$ be a system of word equations. Then the set of solutions of $E$ is identical (in canonical bijection resp.) with the set of solutions of the following equation:*

$$L_1 a \cdots L_k a \, R_1 \cdots R_k \, L_1 b \cdots L_k b \quad = \quad R_1 a \cdots R_k a \, L_1 \cdots L_k \, R_1 b \cdots R_k b.$$

Sometimes it is useful to do the opposite of the proposition above and to split a single word equation into a system where all equations are of type $xy = z$ with $x, y, z \in A \cup \Omega$. This can be derived from the next proposition. Again its (simple) proof is left to the reader.

**Proposition 2.10** *Let $x_1 \cdots x_g = x_{g+1} \cdots x_d$ be a word equation with $1 \leq g < d$, $x_i \in A \cup \Omega$ for $1 \leq i \leq d$. Then the set of solutions of $L = R$ is in canonical bijection with the set of solutions of the following system:*

$$\begin{array}{rclcrcl} x_1 & = & y_1, & \qquad & x_{g+1} & = & y_{g+1}, \\ y_1 x_2 & = & y_2, & & y_{g+1} x_{g+2} & = & y_{g+2}, \\ & \vdots & & & & \vdots & \\ y_{g-1} x_g & = & y_g, & & y_{d-1} x_d & = & y_d, \end{array}$$

$$y_g = y_d.$$

*In the system above $y_1, \ldots, y_d$ denote new variables.*

### 2.5 A single variable

A parametric description of the set of all solutions can be computed in polynomial time, if there is only one variable occurring in the equation. This serves as an example of why $p$-stable normal forms might be useful, but it is not used elsewhere. The reader may skip this subsection.

Let $E$ be a set of word equations where exactly one variable $x$ occurs, $\Omega = \{x\}$. By Prop. 2.9 $E$ is given by a single equation $L = R$ with $L, R \in (A \cup \{x\})^*$. The first check is whether $\sigma(x) = 1$ yields the singular solution. It is then enough to consider only non-singular solutions. Let us denote by $\mathcal{L}$ a list of pairs $(p, r)$ where $p \in A^+$ is primitive and $r \in A^*$ is a prefix $r < p$. We say that $\mathcal{L}$ is *complete* for the equation $L = R$, if every non-singular solution $\sigma$ has the form $\sigma(x) = p^\alpha r$ for some $\alpha \geq 0$ and $(p, r) \in \mathcal{L}$.

Since our intention here is to give an application for $p$-stable normal forms, assume for a moment that a finite complete list $\mathcal{L}$ has already been computed in a first phase of the algorithm. Then we proceed as follows. For each pair $(p, r) \in \mathcal{L}$ we make a first test whether $\sigma(x) = r$ is a solution and a second test whether $\sigma(x) = pr$ is a solution. After that we search (for this pair $(p, r)$) for solutions where $\sigma(x) = p^\alpha r$ with $\alpha \geq 2$. Replace all occurrences of $x$ in the equation $L = R$ by the expression $pp^{\alpha-2}pr$, where $\alpha$ now denotes an integer variable. Thus, the problem is now to find solutions for $\alpha$ such that $\alpha \geq 2$. Using the symbolic expression we can factorize $L$ and $R$ in their $p$-stable normal forms:

$$\begin{aligned} L &= u_0 p^{m_1\alpha+n_1} u_1 \cdots p^{m_k\alpha+n_k} u_k, \\ R &= v_0 p^{m'_1\alpha+n'_1} v_1 \cdots p^{m'_\ell\alpha+n'_\ell} v_\ell. \end{aligned}$$

Note that $k, \ell \geq 0$ and $m_i, m'_j \in \mathbb{N}$, $n_i, n'_j \in \mathbb{Z}$ for $1 \leq i \leq k$ and $1 \leq j \leq \ell$. By Prop. 2.7 we have to verify $k = \ell$, $u_i = v_i$ for $0 \leq i \leq k$ and we have to solve a linear Diophantine system:

$$(m_i \Leftrightarrow m'_i)\alpha = n'_i \Leftrightarrow n_i \qquad \text{for } 1 \leq i \leq k.$$

There are three cases. Either no or exactly one $\alpha \geq 2$ or all $\alpha \geq 2$ satisfy these equations.

Note that for each pair $(p, r)$ the necessary computations can be done in linear time. The performance of the algorithm depends therefore on an efficient computation of a short and complete list $\mathcal{L}$.

We may assume that $L = ux \cdots$ and $R = xvcldots$, where $u \in A^+, v \in A^*$ and both words $u$ and $v$ are of maximal length. Let $p \in A^+$ be the primitive root of $u$, i.e., $p$ is primitive and $u = p^e$ for some $e \geq 1$. If $\sigma$ is a solution of $L = R$, then $\sigma$ solves also an equation of type $ux = xw$ for some word $w \in A^+$. By Prop. 2.1 it is immediate that we have $\sigma(x) = p^\alpha r$ for some $\alpha \geq 0$ and $r < p$. Thus, the simple method is to define the list $\mathcal{L}$ by all pairs $(p, r)$ where $r < p$. We obtain a list $\mathcal{L}$ with $\|p\|$ elements. It is clear that all computations can be done in polynomial time. In fact square time is enough.

There is an improvement to an $\mathcal{O}(\|E\| \log \|E\|)$-algorithm due to Eyono Obono, Goralcik, and Maksimenko (1994). This improvement is a clever method to compute a complete list $\mathcal{L}$ of at most logarithmic length. The method uses a finer combinatorial analysis and it relies, in particular, on the following facts which can be found in Lothaire (1983):

- Let $r, s \in A^*$. If the word $sr$ is primitive, then $rs$ is also primitive.
- Let $p, q \in A^+$ be primitive words and $u = p^e, w = q^f$ for some $e, f \geq 1$. If $u$ and $w$ are conjugates, then $p$ and $q$ are conjugates and there is a unique factorization $p = rs, q = sr$ with $r < p$. Moreover, if $ux = xw$ for some word $x \in A^*$, then we have $x = p^\alpha r$ for some $\alpha \geq 0$ and the unique prefix $r < p$ above.
- Let $p, q, r \in A^+$ be primitive words such that $p^2 < q^2 < r^2$. Then we have $|p| + |q| \leq |r|$. In particular, a word $w \in A^*$ of length $n$ has at most $\mathcal{O}(\log n)$ distinct prefixes of the form $pp$ where $p$ is primitive.

The aim is to compute a complete list $\mathcal{L}$ for the equation $L = R$ of length $\mathcal{O}(|LR|)$. For this purpose we divide the set of non-singular solutions into two classes. The first class contains all solutions where $|\sigma(x)| \geq |u| \Leftrightarrow |v|$. (Of course, in the case $|u| \leq |v|$ all solutions satisfy this condition.) Let $w$ be the prefix of the word $vu$ such that $|w| = |u|$. If $\sigma$ is a solution with $|\sigma(x)| \geq |u| \Leftrightarrow |v|$, then we have $u\sigma(x) = \sigma(x)w$. Let $p$ be the primitive root of $u$ and let $q$ be the primitive root of $w$. Then $\sigma(x) = p^{\alpha}r$ for some $\alpha \geq 0$ and the unique prefix $r < p$ such that $p = rs$ and $q = sr$. If $p$ and $q$ are not conjugates, then there is no such solution. Otherwise, if $p$ and $q$ are conjugates, we include the unique pair $(p, r)$ into $\mathcal{L}$. This pair covers all solutions where $|\sigma(x)| \geq |u| \Leftrightarrow |v|$.

Now, let $\sigma$ be a non-singular solution such that $0 \neq |\sigma(x)| < |u| \Leftrightarrow |v|$. This implies that $R$ has the form $R = xvx \cdots$ and that $\sigma(x)v\sigma(x) < u\sigma(x)$. Hence $\sigma(x)v\sigma(x) < uu$ and $ww < vuu$, where $w$ denotes the non-empty word $v\sigma(x)$. Let $q$ be the primitive root of $w$, then we have $qq < vuu$.

There is a unique factorization $q = sr$ with $s < q$ such that $v \in q^*s$. the word $rs$ is also primitive and we have $\sigma(x) = (rs)^{\alpha}r$ for some $\alpha \geq 0$. Therefore it is enough to compute the list of all primitive words $q$ such that $qq < vuu$. If $v = 1$, then we add all pairs $(q, 1)$ to $\mathcal{L}$. Otherwise, if $v \neq 1$, then we compute for each $q$ the unique factorization $q = sr$ with $s \neq 1$ such that $v \in q^*s$. We add all pairs $(rs, r)$ to $\mathcal{L}$.

## 3 Linear Diophantine equations: Bounding the exponent of periodicity

The input for the algorithm is an equation $L = R$ with $L, R \in (A \cup \Omega)^*$ together with regular languages $L_x \subseteq A^*$ for all variables $x \in \Omega$. We are looking for a solution $\sigma : \Omega \to A^*$ such that $\sigma(L) = \sigma(R)$ and $\sigma(x) \in L_x$ for all $x \in \Omega$. For notational convenience we don't distinguish variables from constants in the equation henceforth. Every constant $a \in A$ is replaced by a new variable $x_a$ and the constraint $L_{x_a} = \{a\}$ for all $a \in A$. (For readability we use constants in examples however.) From now on the equation is given as

$$x_1 \cdots x_g = x_{g+1} \cdots x_d$$

with $x_i \in \Omega$. In order to exclude trivial cases we shall assume $1 \leq g < d$ whenever convenient. The number $d$ is called the *denotational length* of the equation. It is enough to consider non-singular solutions. Hence we shall assume that $1 \notin L_x$ for all $x \in \Omega$. Next we fix a finite semigroup $S$ and a semigroup homomorphism $\varphi : A^+ \to S$ such that $L_x = \varphi^{-1}\varphi(L_x)$ for all $x \in \Omega$. For later use we demand that $\varphi$ is surjective. The semigroup $S$ can be realized as the image $\varphi(A^+)$ of the canonical homomorphism to the direct product of the syntactical monoids with respect to $L_x$ for $x \in \Omega$. Sometimes it is more convenient to work with monoids instead of semigroups. We denote by $S^1$ the monoid, which is obtained by adjoining by a unit element 1 to $S$. We have $S^1 \setminus \{1\} = S$ and the homomorphism $\varphi$ is extended to a monoid homomorphism $\varphi : A^* \to S^1$. We have $\varphi^{-1}(1) = \{1\}$ and $\varphi(A^+) = S$.

Given $S$ we can compute constants $t(S) \geq 0$ and $q(S) > 0$ such that $s^{t(S)+q(S)} = s^{t(S)}$ for all $s \in S^1$. In the following we actually use another constant $c(S)$, which is defined as the least multiple of $q(S)$ such that $c(S) \geq \max\{2, t(S)\}$. Note that this implies $s^{r+\alpha c(S)} = s^{r+\beta c(S)}$ for all $s \in S^1$ and $r \geq 0$ and $\alpha, \beta \geq 1$.

**Remark 3.1** *Assume that each regular language $L_x$ is specified by an NFA with $r_x$ states, $x \in \Omega$. Let $r = \sum_{x\in\Omega} r_x$. Then we may choose the semigroup $S$ such that*

$$|S| \leq 2^{r^2} \textit{ and } c(S) \leq r!.$$

*A proof for these bounds can be found in Markowsky (1977), where a more precise analysis is given. For the moment explicit upper bounds for $|S|$ and $c(S)$ are not used. They are used only later (Sect. 5.3) when complexity issues are investigated.*

The important point here is that our knowledge from linear algebra how to find all minimal solutions of a system of linear Diophantine equations yields an effective upper bound for the exponent of periodicity of a solution of minimal length of a given word equation (with regular constraints). Any effective upper bound would be sufficient, but one can do better. The upper bound is exponential in the input size, and this is essentially optimal. In the proof below a rather detailed analysis is given. So the proof becomes quite technical, which might hide the beautiful and simple idea behind it. Readers who are mainly interested in the pure decidability result are invited to ignore the exact values.

**Theorem 3.2** *Let $d \geq 1$ be a natural number, $\varphi : A^* \to S^1$ a homomorphism, and $c(S) \geq 2$ as above. There is a computable number $e(c(S), d) \in c(S) \cdot 2^{\mathcal{O}(d)}$ satisfying the following assertion.*

*Given as instance a word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$ of denotational length $d$ together with a solution $\sigma' : \Omega \to A^*$, we can effectively find a solution $\sigma : \Omega \to A^*$ and a word $w \in A^*$ such that the following conditions hold:*

1. *$\varphi\sigma'(x) = \varphi\sigma(x)$ for all $x \in \Omega$,*
2. *$w = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$,*
3. *$\exp(w) \leq e(c(S), d)$.*

*Proof*. For $g = 0$ or $g = d$, we have $\exp(w) = 0$, hence let $1 \leq g < d$.

Testing all words of length up to $|\sigma'(x_1 \cdots x_g)|$ we find a solution $\sigma$ and a word $w$ such that $w = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$ is of minimal length among all solutions $\sigma$ where $\varphi\sigma'(x) = \varphi\sigma(x)$ for all $x \in \Omega$. Recall that $x_1 \cdots x_g = x_{g+1} \cdots x_d$ is equivalent to the following system:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
x_1 & = & y_1, & x_{g+1} & = & y_{g+1}, \\
y_1 x_2 & = & y_2, & y_{g+1} x_{g+2} & = & y_{g+2}, \\
& \vdots & & & \vdots & \\
y_{g-1} x_g & = & y_g, & y_{d-1} x_d & = & y_d,
\end{array}$$
$$y_g = y_d$$

Note also that $\exp(w) = \exp(\sigma(y_g))$. After an obvious elimination of variables, the system above is equivalent to a system of $d \Leftrightarrow 2$ equations of type

$$xy = z\,, \qquad x, y, z \in \Omega.$$

Choose a primitive word $p \in A^+$ such that $w = up^{\exp(w)}v$ for some $u, v \in A^*$. Consider an equation $xy = z$ from the system above and write the words $\sigma(x), \sigma(y), \sigma(z)$ in their $p$-stable normal forms:

$$\begin{array}{rcl}
\sigma(x) & : & (u_0, r_1 + \alpha_1 c(S), u_1, \dots, r_k + \alpha_k c(S), u_k), \\
\sigma(y) & : & (v_0, s_1 + \beta_1 c(S), v_1, \dots, s_\ell + \beta_\ell c(S), v_\ell), \\
\sigma(z) & : & (w_0, t_1 + \gamma_1 c(S), w_1, \dots, t_m + \gamma_m c(S), w_m).
\end{array}$$

The natural numbers $r_i, s_i, t_i, \alpha_i, \beta_i$, and $\gamma_i$ are uniquely determined by $w$, $c(S)$, and the requirement $0 \le r_i, s_i, t_i < c(S)$.

Since $w$ is a solution, there are many equations among the words and among the integers. For example, for $k, \ell \ge 2$ we have $u_0 = w_0$, $v_l = w_m$, $r_1 = t_1$, $\alpha_1 = \gamma_1$, etc. In order to be precise, we shall use:

$$\begin{array}{rclcrcl}
\alpha_1 & = & \gamma_1, & \dots, & \alpha_{k-1} & = & \gamma_{k-1}, \\
\beta_2 & = & \gamma_{m-\ell+2}, & \dots, & \beta_\ell & = & \gamma_m.
\end{array}$$

We have no bound on $k$, $\ell$, or $m$, but we have $|k + \ell \Leftrightarrow m| \le 2$. What exactly happens depends on the $p$-stable normal form of the product $u_k v_0$. Since $u_k, v_0 \notin A^* p^2 A^*$, it is enough to distinguish nine cases. Here are the nine possible $p$-stable normal forms of $u_k v_0$, where $t \in \{0, 1\}, u_k, v_0 \in A^*$, and $u'_k, v'_0, w' \in A^+$:

$$\begin{array}{lll}
(u_k v_0), & (p, t, p), & (p, t, pv'_0), \\
(u'_k p, t, p), & (u'_k p, t, pv'_0), & (p, 0, w', 0, p), \\
(p, 0, w', 0, pv'_0), & (u'_k p, 0, w', 0, p), & (u'_k p, 0, w', 0, pv'_0).
\end{array}$$

The case $(p, 0, w', 0, p)$ can be produced, if $p$ has an overlap as in $p = ababa$. Then we might have $u_k = pabab, v_0 = abap$, which yields $u_k v_0 = ppbap = pabpp$ and $abp = pba$. Hence the $p$-stable normal form $u_k v_0$ is $(p, 0, abp, 0, p)$. We may conclude $w_{k+1} = abp$ and

$$t_k + \gamma_k c(S) = r_k + \alpha_k c(S) + 1, \quad t_{k+1} + \gamma_{k+1} c(S) = s_1 + \beta_1 c(S) + 1.$$

In particular $k + \ell = m$. If $r_k < c(S) \Leftrightarrow 1$, then $\alpha_k = \gamma_k$, otherwise $\alpha_k + 1 = \gamma_k$. Similarly, if $s_1 < c(S) \Leftrightarrow 1$, then $\beta_1 = \gamma_{k+1}$, otherwise $\beta_1 + 1 = \gamma_{k+1}$.

A $p$-stable normal form of type $(u'p, 0, w', 0, pv')$ with $u', v', w' \in A^+$ leads to $k + \ell = m + 2$ and $0 = \gamma_k = \gamma_{k+1}$. Let us consider another example. If $u_k v_0 = p^3$, then $k + \ell = m + 1$ and we have

$$r_k + s_1 + 3 + (\alpha_k + \beta_1) c(S) = t_k + \gamma_k c(S).$$

Since by assumption $c(S) \ge 2$, the case $u_k v_0 = p^3$ leads to the equation:

$$\gamma_k \Leftrightarrow (\alpha_k + \beta_1) = c \text{ with } c \in \{0, 1, 2\}.$$

We have seen that there are various possibilities for $u_k v_0$. However, always the same phenomenon arises. First of all we obtain a bunch of trivial equations which can be eliminated by renaming. All equations of type $\gamma = 0$ are eliminated by substitution. Then, for each $xy = z$ either there are at most two equations of type $\gamma = \alpha + 1$ or there is one equation of type $\gamma \Leftrightarrow (\alpha + \beta) = c$ with $c \in \{0, 1, 2\}$. If there are two equations of type $\gamma = \alpha + 1$, then one of them is eliminated by substitution. So after renaming and substituting we end up with at most one non-trivial equation having at most three variables. Proceeding this way through all $d \Leftrightarrow 2$ word equations we have various interactions due to renaming and substitution. However, finally each equation $xy = z$ leads to at most one non-trivial equation with at most three variables. The type of this equation is:

$$\epsilon_1 \gamma + i_1 \Leftrightarrow \epsilon_2 \alpha \Leftrightarrow i_2 \Leftrightarrow \epsilon_3 \beta \Leftrightarrow i_3 = c$$

where we have $0 \leq i_1, i_2, i_3 \leq d \Leftrightarrow 2,\ 0 \leq c \leq 2,\ \epsilon_1, \epsilon_2, \epsilon_3 \in \{0, 1\}$. This can be written as:

$$\epsilon_1 \gamma \Leftrightarrow \epsilon_2 \alpha \Leftrightarrow \epsilon_3 \beta = c' \text{ with } |c'| \leq 2d \Leftrightarrow 2.$$

This type introduces a coefficient $\Leftrightarrow 2$ for $\alpha = \beta$ and $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1$.

We have viewed the symbols $\alpha, \beta, \ldots$ as variables ranging over natural numbers. Going back to the solution $\sigma$ the symbols $\alpha_1, \ldots, \alpha_l, \beta_1, \ldots, \beta_\ell, \gamma_1, \ldots, \gamma_m$ represent concrete values which are given by the word $w$. Some of them might still be zero. These are eliminated now. The reason is that they cannot be replaced by other values without risk of changing the image by $\varphi$. If $\delta \geq 1$ is a remaining value, i.e., a number greater than zero, then we replace it by $\delta = 1 + Z_\delta$ where now $Z_\delta$ denotes a variable over $\mathbb{N}$. For example an equation

$$\gamma \Leftrightarrow \alpha \Leftrightarrow \beta = c'$$

with $\alpha, \beta, \gamma \geq 1$ is transformed to a linear Diophantine equation with integer variables $Z_\alpha, Z_\beta, Z_\gamma \geq 0$ as follows:

$$Z_\gamma \Leftrightarrow Z_\alpha \Leftrightarrow Z_\beta = c' + 1 \text{ with } |c' + 1| \leq 2d \Leftrightarrow 1.$$

Putting all equations of type $xy = z$ together we obtain a (perhaps) huge system of linear equations. After substitution and elimination of variables, we end up with a system of at most $d \Leftrightarrow 2$ equations and $n$ integer variables with $n \leq 3(d \Leftrightarrow 2)$. The absolute values of the coefficients are bounded by 2 and that of the constants by $2d \Leftrightarrow 1$. For each equation the sum over the squares over the coefficients is bounded by 5. The linear Diophantine system is defined by $w$ and the word $w$ provides a non-negative integer solution.

What becomes crucial now is the converse: Every solution in non-negative integers yields by backward substitution a word $w''$ and a solution $\sigma'' : \Omega \to A^*$ satisfying $(i)$ and $(ii)$ of the theorem. Therefore: Since $w$ was chosen of minimal length, the solution of the integer system given by $w$ is a minimal solution with respect to the natural partial ordering of $\mathbb{N}^n$. In this ordering we have $(\alpha_1, \ldots, \alpha_n) \leq (\beta_1, \ldots, \beta_n)$ if and only if $\alpha_i \leq \beta_i$ for all $1 \leq i \leq n$.

For $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ let $\|\vec{\alpha}\| = \max\{\alpha_i \mid 1 \leq i \leq n\}$. All we need is a recursive bound for the following value:

$$e(d) = \max\{\|\vec{\alpha}\| \mid \begin{array}{l} \vec{\alpha} \text{ is a minimal solution of a system of linear Diophantine} \\ \text{equations with at most } d \Leftrightarrow 2 \text{ equations, } 3(d \Leftrightarrow 2) \text{ variables,} \\ \text{where the absolute value of the coefficients is bounded} \\ \text{by 2, the sum over the squares over the coefficients in} \\ \text{each equation is bounded by 5, and the absolute value} \\ \text{of constants is bounded by } 2d \Leftrightarrow 1\}. \end{array}$$

Obviously, there are only finitely many systems of linear Diophantine equations where the number of equations, variables, and the absolute value of coefficients and constants is bounded. For each system the set of minimal solutions is finite, this is a special case of Lemma A in Dickson (1913). Moreover the set of minimal solutions is effectively computable. Hence, the set of values of $\|\vec{\alpha}\|$ above is finite and effectively computable. Therefore $e(d)$ is computable. Since $e(d) + d \Leftrightarrow 1 \geq \alpha_1, \ldots, \beta_1, \ldots$ for original values under the consideration above, we obtain a recursive upper bound for the exponent of periodicity. A much more precise statement is possible. It follows from the work of Kościelski and Pacholski (1996) that $e(d) \in 2^{\mathcal{O}(d)}$, see also Rem. 3.3 below. Hence we can state:

$$\exp(w) \leq 2 + (c(S) \Leftrightarrow 1) + (e(d) + d \Leftrightarrow 1) \cdot c(S) \in c(S) \cdot 2^{\mathcal{O}(d)}.$$

This proves the theorem. □

**Remark 3.3** *The analysis of Kościelski and Pacholski (1996) is more accurate than the one presented here, and it leads to linear Diophantine systems having a slightly different structure. The authors make use of the results given in the paper von zur Gathen and Sieveking (1978) They show that the exponent of periodicity of a minimal solution of a word equation of denotational length $d$ is in $\mathcal{O}(2^{1.07d})$. The authors don't consider regular constraints, but, as it is shown above, this doesn't change the situation very much: It yields the factor $c(S)$. Therefore the actual result including regular constraints is:*

$$e(c(S), d) \in c(S) \cdot \mathcal{O}(2^{1.07d}).$$

*It is rather difficult to obtain this very good bound. However, we can circumvent this difficulty. A bound which is already nice and certainly good enough to establish Thm. 3.2 is $e(d) \in \mathcal{O}(2^{cd})$ for some constant $c$, say $c = 4$. Such a more moderate bound can be obtained without any difficulty using the present approach and some standard knowledge in linear algebra, see Prob. 3.1 below.*

**Example 3.4** *Consider $c, n \geq 2$ and let $S = \mathbb{Z}/c\mathbb{Z}$ be the cyclic group of $c$ elements. We give a regular constraint for the variable $x_1$ by defining*

$$L_{x_1} = \{w \in A^+ \mid |w| \equiv 0 \pmod{c}\}.$$

*The system is given by*

$$x_1 = a^c, \quad x_2 = x_1^2, \quad \ldots, \quad x_n = x_{n-1}^2.$$

*Its unique solution $\sigma$ is: $\sigma(x_i) = a^{c \cdot 2^{i-1}}$, $1 \leq i \leq n$. A transformation into a single equation according to Prop. 2.9 shows that $e(c(S), d) \in c(S) \cdot 2^{\Omega(d)}$. Thus, the assertion given in Thm. 3.2 is essentially optimal.*

The following example shows that the length of a minimal solution can be very long although the exponent of periodicity is bounded by a constant.

**Example 3.5** *Let $n \geq 1$. Consider the following well-known system of word equations:*

$$\begin{array}{rclcrcl} x_0 & = & a, & \quad & y_0 & = & b, \\ x_i & = & x_{i-1}y_{i-1}, & \quad & y_i & = & y_{i-1}x_{i-1} \text{ for } 1 \leq i \leq n. \end{array}$$

*The unique solution is the Thue-Morse word:*

$$\sigma(x_n) = abbabaabbaababbabaababbaabbabaab \cdots \text{ for } n \geq 5.$$

*We have $|\sigma(x_n)| = 2^n$, but $\exp(\sigma(x_n)) = 2$.*

**Example 3.6** *Consider the equation with regular constraints:*

$$axyz = zxay,$$

$$L_x = a^2 a^*, \quad L_y = \{a,b\}^* \setminus (a^* \cup b^*), \quad L_z = \{a,b\}^+.$$

*A suitable homomorphism $\varphi : \{a,b\}^+ \to S$ is given by the canonical homomorphism onto the quotient semigroup of $\{a,b\}^+$, which is presented by the defining relations*

$$a^2 = a^3, b = b^2, ab = ba = aab.$$

*Thus, $S$ is a semigroup with a zero, $0 = ab$; and $S$ has four elements:*

$$S = \{a, a^2, b, 0\}.$$

*The constant $c(S) = 2$ fits the requirement $s^{r+c(S)} = s^{r+\alpha c(S)}$ for all $s \in S^1$ and $r \geq 0$, $\alpha \geq 1$. It is not difficult to find a solution $\sigma$ for the equation above, e.g. $\sigma(x) = a^2, \sigma(y) = ba^2$, and $\sigma(z) = a^3ba^2$. Now let $\alpha$, $\beta$, $\gamma$, $\delta$ be some integer variables and let $u, v$, and $w$ be parametric words, which are described by the following a-stable normal forms:*

$$u : (a, 2\alpha, a), \quad v : (ba, 2\beta, a), \quad w : (a, 1+2\gamma, aba, 2\delta, a).$$

*In order to derive the system of linear Diophantine equations, we make a direct approach: We want to solve $auvw = wuav$. First we write $auvw$ as a sequence of a–stable normal forms:*

$$((a), (a, 2\alpha, a), (ba, 2\beta, a), (a, 1+2\gamma, aba, 2\delta, a)).$$

*The resulting a–stable normal form is:*

$$(a, 2\alpha+1, aba, 2\beta+2\gamma+3, aba, 2\delta, a).$$

*Now consider the right-hand side $wuav$. This yields:*

$$(a, 2\gamma+1, aba, 2\alpha+2\delta+3, aba, 2\beta, a).$$

*We obtain the linear Diophantine system:*

$$\begin{aligned} 2\alpha+1 &= 2\gamma+1, \\ 2\beta+2\gamma+3 &= 2\alpha+2\delta+3, \\ 2\delta &= 2\beta. \end{aligned}$$

*Going back to the equation we see that for all $\alpha \geq 0$ and $\beta \geq \alpha$ the mapping*

$$\sigma(x) = a^{2+2\alpha}, \quad \sigma(y) = ba^{2+2\beta}, \quad \sigma(z) = a^{3+2\alpha}ba^{2+2\beta}$$

*yields a solution of the equation $axyz = zxay$ satisfying the regular constraints.*

# 4 Boundary equations

## 4.1 Linear orders over $S$

Let us start with an informal explanation of the notions discussed in this subsection. Assume that $x_1 \cdots x_g = x_{g+1} \cdots x_d\,, 1 \leq g < d$, $x_i \in \Omega$ for $1 \leq i \leq d$ is a solvable word equation with regular constraints and that there is a non-singular solution $\sigma$. The solution is given by some word $w \in A^+$. The equation corresponds to two factorizations $w = u_1 \cdots u_g = u_{g+1} \cdots u_d$ yielding two sequences of non-empty words:

$$(u_1, \dots , u_g), \qquad (u_{g+1}, \dots , u_d).$$

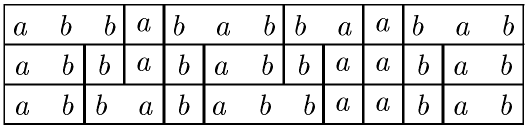
Using the word $w$ these sequences can be merged into a single one such that each $u_i$ is a product of some $w_k$:

$$w = (w_1, \dots , w_m), \quad w_k \neq 1,\ 1 \leq k \leq m,\ m < d.$$

Let us see what happens if we pass via $\varphi$ to the finite semigroup $S$. Two sequences $(p_1, \dots , p_g) \in S^g$ and $(p_{g+1}, \dots , p_d) \in S^{d-g}$ are merged into a single sequence $(s_1, \dots , s_m) \in S^m, m < d$ such that each $p_i \in S$ is a product of some $s_k$. We shall say that $(s_1, \dots , s_m)$ is a *common refinement* of $(p_1, \dots , p_g)$ and $(p_{g+1}, \dots , p_d)$. However, for a given input $d$, there are only finitely many possibilities for sequences of the form $(s_1, \dots , s_m),\ s_j \in S, 1 \leq j \leq m < d$. Thus, in a non-deterministic step we can guess and fix such a sequence which is the $\varphi$-image of $(w_1, \dots , w_m)$.

A basic technique of solving word equations is to split a variable. Working over the sequence $(s_1, \dots , s_m) \in S^m$, a splitting of a variable $x = x'x''$ is accompanied with a splitting of some $s_i$ and a guess of $s', s'' \in S$ such that $s_i = s's''$. In this way the length of the sequences is increasing.

**Example 4.1** *Consider the equation $xauzau = yzbxaaby$. The solution, which was given in Ex. 1.1, leads to the sequences $(abb, a, bab, ba, a, bab)$ and $(ab, ba, b, abb, a, a, b, ab)$, where $(ab, b, a, b, ab, b, a, a, b, ab)$ is a common refinement. This can be best visualized by the following figure.*



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a$ | $b$ | $b$ | $a$ | $b$ | $a$ | $b$ | $b$ | $a$ | $a$ | $b$ | $a$ | $b$ |
| $a$ | $b$ | $b$ | $a$ | $b$ | $a$ | $b$ | $b$ | $a$ | $a$ | $b$ | $a$ | $b$ |
| $a$ | $b$ | $b$ | $a$ | $b$ | $a$ | $b$ | $b$ | $a$ | $a$ | $b$ | $a$ | $b$ |

*Passing to the semigroup $S$ from Ex. 3.6 we could start to search for a solution with the sequence $(0, b, a, b, 0, b, a, a, b, 0) \in S^{10}$.*

We now start the formal discussion of this section. The semigroup $S$ and the homomorphism $\varphi : A^+ \twoheadleftrightarrow S$ is given as in precedent section. An *$S$-sequence* is a sequence $(s_1, \dots , s_m) \in S^m$, $m \geq 0$. A *representation* of $(s_1, \dots , s_m)$ is a triple $(I, \leq, \varphi_I)$ such that $(I, \leq)$ is a totally ordered set of $m + 1$ elements and

$$\varphi_I : \{(i, j) \in I \times I \mid i \leq j\} \to S^1$$

is a mapping satisfying for some order respecting bijection $\rho : I \xrightarrow{\sim} \{0, \ldots , m\}$ the condition

$$\varphi_I(i, j) = s_{\rho(i)+1} \cdots s_{\rho(j)} \in S^1 \text{ for all } i, j \in I, i \leq j.$$

Note that we have $\varphi_I(i, j) = 1$ if and only if $i = j$ and $\varphi_I(i, k) = \varphi_I(i, j)\varphi_I(j, k)$ for all $i, j, k \in I, i \leq j \leq k$.

The *standard representation* of $(s_1, \ldots , s_m)$ is simply $(I, \leq, \varphi_I)$ where $I = \{0, \ldots , m\}$ and $\varphi_I(i, j) = s_{i+1} \cdots s_j$ for $i, j \in I, i \leq j$. Hence for the standard representation the bijection $\rho$ is the identity.

In the following any representation $(I, \leq, \varphi_I)$ of some $S$-sequence is called a *linear order over $S$*.

**Remark 4.2** *An $S$-sequence can be viewed as an abstraction of a linear order over $S$. In most cases we are interested in the abstract objects only, but if we work with them we have to pass to concrete representations. When counting linear orders over $S$ (c.f. Lem. 4.9 below), by convention, we count only standard representations and mappings between them.*

Let $w = a_1 \cdots a_m \in A^*, a_i \in A$ for $1 \leq i \leq m$. The set $\{0, \ldots , m\}$ is the *set of positions* of $w$, and for $0 \leq i \leq j \leq m$ let $w(i, j)$ denote the factor $a_{i+1} \cdots a_j$. The associated $S$-sequence of $w$ is defined by $w_S = (\varphi(a_1), \ldots , \varphi(a_m))$. The notation $w_S$ refers also to its standard representation $w_S = (\{0, \ldots , m\}, \leq, \varphi_w)$. The mapping $\varphi_w$ is defined by $\varphi_w(i, j) = \varphi(w(i, j))$ for all $0 \leq i \leq j \leq m$.

**Definition 4.3** *Let $s, s'$ be $S$-sequences given by some representations $(I, \leq, \varphi_I)$ and $(I', \leq, \varphi_{I'})$. We say that $s'$ is a* refinement *of $s$ (or that $s$* matches *$s'$), if there exists an order respecting injective mapping $\rho : I \to I'$ such that $\varphi_I(i, j) = \varphi_{I'}(\rho(i), \rho(j))$ for all $i, j \in I, i \leq j$. We write either $s \leq s'$ or, more precisely, $s \leq_\rho s'$ and $(I, \leq, \varphi_I) \leq_\rho (I', \leq, \varphi_{I'})$ in this case.*

**Remark 4.4** *Let $s, s'$ be $S$-sequences such that $s \leq s'$. Then we may choose concrete representations and a refinement $(I, \leq, \varphi_I) \leq_\rho (I', \leq, \varphi'_I)$ such that $\rho : I \to I'$ is an inclusion, i.e., $I \subseteq I'$ and $\varphi_I$ is the restriction of $\varphi_{I'}$ to $I$.*

**Definition 4.5** *Let $s$ be an $S$-sequence and $(I, \leq, \varphi_I)$ some representation. A word $w \in A^*$ is called* model *of $s$ (of $(I, \leq, \varphi_I)$ resp.), if the associated $S$-sequence $w_S$ is a refinement of $s$, i.e. , $(I, \leq, \varphi_I) \leq_\rho w_S$ for some $\rho$.*

If $w$ is a model of $s$, then we write $w \models s$ or $w \models (I, \leq, \varphi_I)$. By abuse of language, we make the following convention. As soon as we have chosen a word $w$ as a model, we are free to view the set $I$ as a subset of positions of $w$, i.e., $\rho$ becomes an inclusion and therefore $\varphi_I(i, j) = \varphi(w(i, j))$ for all $i, j \in I, i \leq j$.

**Lemma 4.6** *Every $S$-sequence $(s_1, \ldots , s_m)$ has a model $w \in A^*$.*

*Proof*. Since $\varphi$ is surjective, there are non-empty words $w_i \in A^+$ such that $s_i = \varphi(w_i)$ for all $1 \leq i \leq m$. Let $w = w_1 \cdots w_m$, then we have $w \models (s_1, \ldots , s_m)$. □

The lemma above will yield the positive termination step in Makanin's algorithm if there are no more variables. In the positive case we can eventually reconstruct some $S$-sequence such that some model $w$ describes a solution of the word equation.

Let $i, j \in I, i \leq j$ be positions in a linear order over $S$. Then $[i,j]$ denotes the interval from $i$ to $j$, this is a linear sub-order over $S$ which is induced by the subset $\{k \in I \mid i \leq k \leq j\}$. More generally, let $T \subseteq I$ be a subset, then we view $(T, \leq, \varphi_T)$ as a linear suborder of $(I, \leq, \varphi_I)$. In the following $\min(T)$ and $\max(T)$ refer to the minimal respectively to the maximal element of a subset $T$ of a linear order $I$.

**Definition 4.7** *Let $(I, \leq, \varphi_I)$ be a representation of some $S$-sequence, $T \subseteq I$ a non-empty subset, and $\ell^*, r^* \in I$ be positions such that $\ell^* < r^*$.*

*An* admissible extension of $(I, \leq, \varphi_I)$ by $T$ at $[\ell^*, r^*]$ *is given by a linear order $(I^*, \leq, \varphi_{I^*})$ and two refinements $(I, \leq, \varphi_I) \leq_\rho (I^*, \leq, \varphi_{I^*})$ and $(T, \leq, \varphi_T) \leq_{\rho^*} (I^*, \leq, \varphi_{I^*})$ such that the following two conditions are satisfied:*

1. $I^* = \rho(I) \cup \rho^*(T)$,
2. $\min(\rho^*(T)) = \ell^*$ *and* $\max(\rho^*(T)) = r^*$.

The intuition behind the last definition it should be rather clear. An admissible extension refines $(I, \leq, \varphi_I)$ by defining new positions between $\ell^*$ and $r^*$ until $T$ matches the enlarged interval $[\ell^*, r^*]$ in such a way that all new points have a corresponding point in $T$ and such that $\min(T)$ is mapped to $\ell^*$ and $\max(T)$ is mapped to $r^*$. The other way round: Let $(I^*, \leq, \varphi_{I^*})$ denote an admissible extension of $(I, \leq, \varphi_I)$ by $T$ at $[\ell^*, r^*]$, then we may view $I \subseteq I^*$, whence $T \subseteq I^*$. There is a subset $T^* \subseteq I^*$ representing the same $S$-sequence as $T$; and we have $I^* = I \cup T^*$, $\min(T^*) = \ell^*$, and $\max(T^*) = r^*$.

**Example 4.8** *Let $(s_1, \ldots, s_6)$ be some $S$-sequence, $(I, \leq, \varphi_I)$ its standard representation, $\ell^* = 4$ and $r^* = 6$. Let $(I^*, \leq, \varphi_{I^*})$ represent an admissible extension of $(I, \leq, \varphi_I)$ by $\{0, 3, 4, 5\}$ at $[4, 6]$. Then we may assume $I^* = \{0, \ldots, 6\} \cup \{3^*, 4^*\}$ with $0 < 1 < 2 < 3 < 4 < 5 < 6$ and $4 < 3^* < 4^* < 6$.*

*We may or may not have $5 \in \{3^*, 4^*\}$. Say we have $5 = 3^*$. Then the corresponding $S$-sequence has the form*

$$(s_1, s_2, s_3, s_4, s_5, s_4, s_5)$$

*such that $s_5 = s_1 s_2 s_3$ and $s_6 = s_4 s_5$.*

**Lemma 4.9** *Given $(I, \leq, \varphi_I), T \subseteq I, \ell^*, r^* \in I$. Then the list of all admissible extensions of $(I, \leq, \varphi_I)$ by $T$ at $[\ell^*, r^*]$ is finite and effectively computable.*

*Proof.* Trivial, since the cardinality of an admissible extension is bounded by $|I| + |T|$. □

## 4.2 From word equations to boundary equations

Let $x_1 \cdots x_g = x_{g+1} \cdots x_d, 1 \leq g < d$, $x_i \in \Omega$ for $1 \leq i \leq d$ be a word equation with regular constraints $L_x \subseteq A^*$ for all $x \in \Omega$. Recall we are only interested in non-singular solutions and that we fixed a homomorphism $\varphi : A^+ \to S$ to a finite semigroup $S$ such that $\varphi^{-1}\varphi(L_x) = L_x$ for $x \in \Omega$. Hence without restriction it holds $1 \notin L_x \neq \emptyset$ for all $x \in \Omega$. Since the images $\varphi(L_x) \subseteq S$ are finite sets we can split into finitely many cases where in each case $\varphi(L_x)$ is a singleton. Thus, it is enough to consider a situation where the input is

$x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \leq g < d$ and the question is the existence of a non-singular solution $\sigma : \Omega \to A^+$ satisfying $\psi = \varphi \circ \sigma$ for some fixed mapping $\psi : \Omega \to S$. The question will be reformulated in terms of boundary equations. A system of boundary equations is defined as follows.

**Definition 4.10** *Let $n \geq 0$ and $\varphi : A^+ \to S$ be a homomorphism to a finite semigroup $S$.*

1. *A* system of boundary equations *is specified by a tuple*

$$\mathcal{B} = ((, ,^{-}), (I, \leq, \varphi_I), \mathit{left}, B)$$

*where , is a set of $2n$ variables, $^{-}$ : , $\to$ , is an involution without fixed points, i.e., $\overline{\overline{x}} = x$, $x \neq \overline{x}$, for all $x \in$ , , the triple $(I, \leq, \varphi_I)$ is a linear order over $S$, $\mathit{left}$ : , $\to I$ is a mapping, and $B$ is a set of* boundary equations*. Every boundary equation $b \in B$ has the form $b = (x, i, \overline{x}, j)$ with $x \in$ , , $i, j \in I$ such that $\mathit{left}(x) \leq i$ and $\mathit{left}(\overline{x}) \leq j$.*

2. *A* solution *of $\mathcal{B}$ is a model $w \models (I, \leq, \varphi_I)$, $w \in A^*$, such that*

$$w(\mathit{left}(x), i) = w(\mathit{left}(\overline{x}), j) \text{ for all } (x, i, \overline{x}, j) \in B.$$

*(Recall that if a word $w \in A^*$ is a model for $(I, \leq, \varphi_I)$, then we view $I$ as a subset of positions of $w$. Hence it makes sense to write $w(p, q)$ for $p, q \in I$, $p \leq q$.)*

3. *If $\mathcal{B}$ is solvable, then the* exponent of periodicity $\exp(\mathcal{B})$ *of $\mathcal{B}$ is defined by*

$$\exp(\mathcal{B}) = \min\{\exp(w) \mid w \text{ is a solution of } \mathcal{B}\}.$$

**Remark 4.11** *If $n = 0$, then there are no variables, hence no boundary equations, and any model $w \models (I, \leq, \varphi_I)$ is a solution is a solution of $\mathcal{B}$. In particular, if $n = 0$, then the system is solvable by Lem. 4.6.*

Consider a word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$ and a mapping $\psi : \Omega \to S$. We are going to construct a system

$$\mathcal{B} = ((, ,^{-}), (I, \leq, \varphi_I), \mathrm{left}, B)$$

of boundary equations having the following two properties.

1.) Let $\sigma : \Omega \to A^+$ be a solution of the word equation such that $\psi = \varphi \circ \sigma$, and let $v \in A^*$ be a word with $v = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$. Then $w = vv$ is a solution of $\mathcal{B}$.

2.) Let $w \models (I, \leq, \varphi_I)$ be a solution of $\mathcal{B}$. Then we have $w \in A^*vvA^*$ for some $v \in A^*$ and there is a solution of the word equation $\sigma : \Omega \to A^+$ such that $\psi = \varphi \circ \sigma$ and $v = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$.

In order to define $\mathcal{B}$ we start with the $S$-sequence

$$(\psi(x_1), \ldots, \psi(x_d)).$$

Let $(I, \leq, \varphi_I)$ be some representation, $I = \{i_0, \ldots, i_d\}$, $i_0 \leq \cdots \leq i_d$. The next step is to define the pair $(, ,^{-})$ and the mapping left : , $\to I$. To this purpose we introduce an undirected graph. Let $(V, E)$ be the undirected graph with vertex set $V = \{1, \ldots, d\}$ and edge set $E = \{(p, q) \in V \times V \mid x_p = x_q\}$. The idea is that for $v = \sigma(x_1, \ldots, x_g) = \sigma(x_{g+1}, \ldots, x_d)$ and $w = vv$ we can realize $I$ as a subset of positions of $w$ such that both $w \models (\psi(x_1), \ldots, \psi(x_d))$ and the following equations hold:

$$w(i_0, i_g) = w(i_g, i_d), \quad w(i_{p-1}, i_p) = w(i_{q-1}, i_q) \text{ for all } (p, q) \in E.$$

For the first equation we shall introduce an extra variable $x_0$ (and its dual $\overline{x_0}$) below; in the other list of equations there is some redundancy. For $(p, q), (q, r) \in E$, we have by definition $(p, r) \in E$, but the equations $w(i_{p-1}, i_p) = w(i_{q-1}, i_q)$ and $w(i_{q-1}, i_q) = w(i_{r-1}, i_r)$ already imply $w(i_{p-1}, i_p) = w(i_{r-1}, i_r)$. Hence we don't need the edge $(p, r)$ for the equation. To avoid this redundancy we let $F \subseteq E$ be a spanning forest of $(V, E)$. This means $F = F^{-1}$, $F^* = E^*$, and $(V, F)$ is an acyclic undirected graph. We have $|F| = 2(d \Leftrightarrow c)$, where $c$ is the number of connected components of $(V, E)$. The elements of $F$ are called variables and for each $x = (p, q) \in F$ we define its dual and two positions left$(x)$, right$(x)$:

$$\overline{x} = (q, p), \ \text{left}(x) = i_{p-1}, \ \text{right}(x) = i_p.$$

Note that $x \neq \overline{x}$ and $\overline{\overline{x}} = x$ for all $x \in F$. Taking duals corresponds to edge reversing in $(V, F)$. Define two new extra variables $x_0$ and $\overline{x_0}$ with $\overline{\overline{x_0}} = x_0$ and define $, = \{x_0, \overline{x_0}\} \cup F$ and:

$$\text{left}(x_0) = i_0, \ \text{right}(x_0) = i_g = \text{left}(\overline{x_0}), \ \text{right}(\overline{x_0}) = i_d.$$

This defines the set , , the involution without fixed points $^- : , \to ,$ , and the mapping left $: , \to I$.

The last step of the construction is to define the set $B$ of boundary equations. It should be clear what to do. We define

$$B = \{(x, \text{right}(x), \overline{x}, \text{right}(\overline{x})) \mid x \in , \}.$$

We have to verify two properties.

1. Let $\sigma : \Omega \to A^+$ be a solution such that $\psi = \varphi \circ \sigma$, and let $w = vv$, where $v = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$. The word $w$ has positions $0 = i_0 < i_1 < \cdots < i_d$, where $i_d$ is the last position and the following equations hold:

   $$w(i_0, i_g) = w(i_g, i_d), \quad w(i_{p-1}, i_p) = \sigma(x_p) \text{ for } 1 \leq p \leq d.$$

   In particular, $w \models (I, \leq, \varphi_I)$ and $w$ is a solution of $\mathcal{B}$.

2. Let $w \models (I, \leq, \varphi_I)$ be a solution of $\mathcal{B}$. Without restriction we may view $I$ as a subset of positions of $w$. Consider the factors $w(i_0, i_g)$ and $w(i_g, i_d)$. The boundary equation $(x_0, \text{right}(x_0), \overline{x_0}, \text{right}(\overline{x_0})) \in B$ implies $w(i_0, i_g) = w(i_g, i_d)$ and it follows that $w \in A^*vvA^*$ for $v = w(i_0, i_g)$. We define $\sigma : \Omega \to A^+$ by $\sigma(x_p) = w(i_{p-1}, i_p)$. Since $i_{p-1} < i_p$, this is a non-empty word. The elements $(x, \text{right}(x), \overline{x}, \text{right}(\overline{x})) \in B$ for $x = (p, q)$, $\overline{x} = (q, p)$, $(p, q) \in T$ imply $w(i_{p-1}, i_p) = w(i_{q-1}, i_q)$ whenever $x_p = x_q$. Hence $\sigma$ is well-defined. We have $\varphi\sigma(x_p) = \varphi w(i_{p-1}, i_p) = \psi(x_p)$ since $w \models (I, \leq, \varphi_I)$. Finally, $v = w(i_0, i_g) = w(i_g, i_d)$ implies $v = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$.

Thus, the word equation with regular constraints given by the mapping $\psi$ has a solution if and only if the system of boundary equations is solvable. The construction of the system $\mathcal{B}$ above can be performed in polynomial time; more precisely, the construction yields a logspace-reduction. Due to this reduction, Makanin's result follows from the next theorem. The assertion of Thm. 4.12 is in fact equivalent to Makanin's result, see Lem. 4.14 below.

**Theorem 4.12** *It is decidable whether a system of boundary equations has a solution.*

The rest of this chapter is devoted to the proof of Thm. 4.12. An important property is stated in the next proposition: We can bound the exponent of periodicity while searching for a solution. Note however that bounding the exponent of periodicity of some word gives absolutely no bound on the length of this word.

**Proposition 4.13** *Given as instance a system of boundary equations $\mathcal{B}$, we can compute a number $e(\mathcal{B})$ having the property that if $\mathcal{B}$ is solvable, then we have* $\exp(\mathcal{B}) \le e(\mathcal{B})$.

The proof of Prop. 4.13 could be based on the same techniques as presented in Sect. 3. However, for our purposes we prefer to prove Prop. 4.13 via a reduction to word equations.

**Lemma 4.14** *There is an effective reduction of the solvability of a system of boundary equations $\mathcal{B}$ to some word equation with regular constraints. Moreover, there is a reduction such that if $w \in A^*$ is a solution of the word equation, then $\mathcal{B}$ is solvable and we have* $\exp(\mathcal{B}) \le \exp(w)$.

*Proof*. Let $\mathcal{B} = ((,\ ,^{-}), (I, \le, \varphi_I), \mathrm{left}, B)$ be a system of boundary equations. We may assume that the linear order $(I, \le, \varphi_I)$ is the standard representation of its underlying $S$-sequence $s = (s_1, \ldots, s_m)$. Introduce new variables $y_1, \ldots, y_m$ with regular constraints $\psi(y_p) = s_p$, $1 \le p \le m$.

For each boundary equation $b = (x, i, \overline{x}, j) \in B$ we introduce a word equation

$$y_{\mathrm{left}(x)+1} \cdots y_i = y_{\mathrm{left}(\overline{x})+1} \cdots y_j.$$

This system of word equations with regular constraints is solvable if and only if $\mathcal{B}$ is solvable. Indeed, if $w \in A^*$ is a solution of $\mathcal{B}$, then, by definition, we have $(I, \le, \varphi_I) \le_\rho w_S$, and $\rho(I)$ is a subset of positions of $w$. All word equations

$$w(\rho(\mathrm{left}(x)), \rho(i)) = w(\rho(\mathrm{left}(\overline{x})), \rho(j))$$

are satisfied for $(x, i, \overline{x}, j) \in B$. Hence defining $\sigma(y_p) = w(\rho(p \Leftrightarrow 1), \rho(p))$, $1 \le p \le m$ yields a solution of the system of word equations.

For the other direction let $\sigma(y_p) = v_p$, $1 \le p \le m$ be some solution of the system of word equations. Due to the regular constraints we have $\psi(y_p) = s_p$ and $v_p \ne 1$ for all $1 \le p \le m$. Therefore the word $v = \sigma(y_1) \cdots \sigma(y_m)$ solves $\mathcal{B}$.

Next, we transform the system of word equations into a single word equation $L = R$ using Prop. 2.9 and finally we reduce to the word equation $Ly_1 \cdots y_m = Ry_1 \cdots y_m$. The point is that if $w$ is a solution of this equation, then some suffix $v$ of $w$ solves $\mathcal{B}$. Hence $\exp(\mathcal{B}) \le \exp(v) \le \exp(w)$. This yields Lem. 4.14. Now, let $d$ be the denotational length of $Ly_1 \cdots y_m = Ry_1 \cdots y_m$. Then define the number $e(\mathcal{B}) = e(c(S), d)$, which has been given in Thm. 3.2. We can choose $w$ such that $\exp(w) \le e(c(S), d)$. This proves Prop. 4.13. □

### 4.3 The convex chain condition

Let $\mathcal{B} = ((\,,\,^{-}), (I, \leq, \varphi_I), \text{left}, B)$ be a system of boundary equations. A boundary equation $b = (x, i, \overline{x}, j) \in B$ is also called a *brick* henceforth. The variable $x$ is called the *label* of the brick $b = (x, i, \overline{x}, j)$. Pictorially a brick is given as follows:

$$\begin{array}{|lr|}\hline x & i \\ \hline \overline{x} & j \\ \hline\end{array}$$

The dual brick $\overline{b}$ of $b = (x, i, \overline{x}, j)$ is given by reversing the brick, it has the label $\overline{x}$:

$$\begin{array}{|lr|}\hline \overline{x} & j \\ \hline x & i \\ \hline\end{array}$$

Henceforth, we make the assumption that $B$ is closed under duals (i.e., $b \in B$ implies $\overline{b} \in B$) and that there is at least one brick $b \in B$ having label $x$ for all $x \in\,$, . Clearly, this is no restriction. For $x \in\,$, let $B(x) \subseteq B$ be the subset of bricks with label $x$. Then $B(x) = \{(x, i_1, \overline{x}, j_1), \ldots, (x, i_r, \overline{x}, j_r)\}$ for some non-empty subset $\{i_1, \ldots, i_r\} \subseteq I$ such that $\text{left}(x) \leq i_1 \leq \cdots \leq i_r$. The *right boundary* of $x$ is defined by by $\text{right}(x) = i_r$.

Before we continue, we make some additional assumptions on $B$. All of them are necessary conditions for solvability and easily verified.

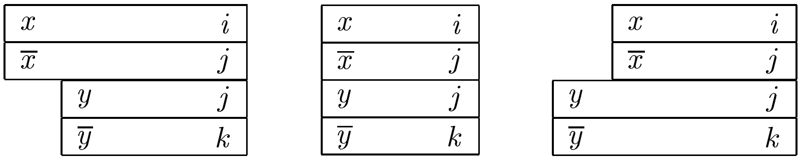
Let $(x, i, \overline{x}, j), (y, i, \overline{y}, j), (y, i', \overline{y}, j') \in B$. Then we assume from now on:

- $\text{left}(x) \leq \text{left}(\overline{x})$ if and only if $i \leq j$,
- $\varphi_I(\text{left}(x), i) = \varphi_I(\text{left}(\overline{x}), j)$,
- $\text{left}(x) \leq \text{left}(y)$ if and only if $\text{left}(\overline{x}) \leq \text{left}(\overline{y})$,
- $i \leq i'$ if and only if $j \leq j'$.

These assumptions imply that if $B(x) = \{(x, i_1, \overline{x}, j_1), \ldots, (x, i_r, \overline{x}, j_r)\}$ is given such that $\text{left}(x) \leq i_1 \leq \cdots \leq i_r$, then we also have $\text{left}(\overline{x}) \leq j_1 \leq \cdots \leq j_r$. In particular, $B(x)$ contains a brick $(x, \text{right}(x), \overline{x}, \text{right}(\overline{x}))$. The set $B(x)$ can be depicted as follows:

$$B(x) = \left\{ \begin{array}{|lr|}\hline x & i_1 \\ \hline \overline{x} & j_1 \\ \hline\end{array}, \begin{array}{|lr|}\hline x & i_2 \\ \hline \overline{x} & j_2 \\ \hline\end{array}, \ldots, \begin{array}{|lr|}\hline x & \text{right}(x) \\ \hline \overline{x} & \text{right}(\overline{x}) \\ \hline\end{array} \right\}$$

In our pictures a brick $(x, i, \overline{x}, j)$ can be placed upon $(y, j', \overline{y}, k)$, if and only if $j = j'$. We obtain one of out of three different shapes:



Which one of these cases occurs is determined by the function left : , $\to I$. The leftmost picture corresponds to $\text{left}(\overline{x}) < \text{left}(y)$, the picture in the middle corresponds to $\text{left}(\overline{x}) = \text{left}(y)$, the picture on the right means $\text{left}(\overline{x}) > \text{left}(y)$.

**Definition 4.15** *Let $m \geq 1$. A* chain $C$ of length $m$ *is a sequence of bricks*

$$C = ((x_1, i_1, \overline{x_1}, i_2), (x_2, i_2, \overline{x_2}, i_3), \ldots, (x_m, i_m, \overline{x_m}, i_{m+1})),$$

*where $(x_p, i_p, \overline{x_p}, i_{p+1}) \in B$ for all $1 \leq p \leq m$.*

*A chain $C$ is called* convex, *if for some index $q$ with $1 \leq q \leq m$ we have:*

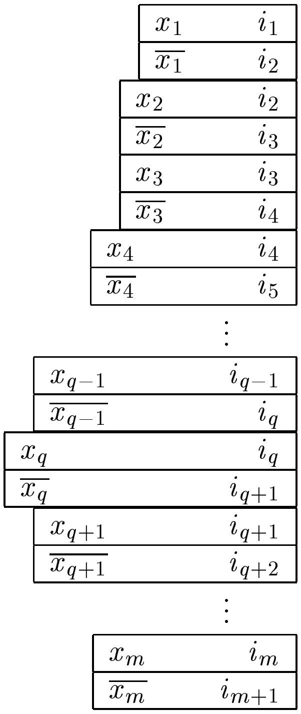
$$\begin{array}{rcll} \mathit{left}(\overline{x_p}) & \geq & \mathit{left}(x_{p+1}) & \mathit{for}\ 1 \leq p < q, \\ \mathit{left}(\overline{x_p}) & \leq & \mathit{left}(x_{p+1}) & \mathit{for}\ q \leq p < m. \end{array}$$

*A convex chain $C$ is called* clean, *if the bricks of $C$ are pairwise distinct.*

*A brick $(x, i, \overline{x}, j)$ is* linked via a convex chain of length $m$ *to the brick $(x', i', \overline{x}', j')$, if there is a convex chain $C$ of length $m$ as above such that $m \geq 1, (x, i, \overline{x}, j) = (x_1, i_1, \overline{x_1}, i_2)$, and $(x', i', \overline{x}', j') = (x_m, i_m, \overline{x_m}, i_{m+1})$.*

**Remark 4.16** *If $C = (b_1, \ldots, b_m)$ is a convex chain, then its dual $\overline{C} = (\overline{b_m}, \ldots, \overline{b_1})$ and $(b_p, \ldots, b_q)$, $1 \leq p \leq q \leq m$ are convex chains. If $b_p = (x_p, i_p, \overline{x_p}, i_p)$ for some $1 < p < m$, then $(b_1, \ldots, b_{p-1}, b_{p+1}, \ldots b_m)$ is a convex chain. If $b_p = b_q$ for some $1 < p < q \leq m$, then $(b_1, \ldots, b_{p-1}, b_q, \ldots b_m)$ is also a convex chain. In particular, if two bricks are linked via a convex chain, then they are linked via some clean convex chain. The shortest chain linking two bricks to each other is certainly clean.*

A typical picture of a convex chain is depicted in the following figure.



**Definition 4.17** *Let $F \subseteq I$ be a subset. A brick $(x, i, \overline{x}, j) \in B$ is called a* basis *or* foundation *with respect to $F$, if $j \in F$. We say that $\mathcal{B}$ satisfies the* convex chain condition *(with respect to $F$), if every brick $b \in B$ can be linked via some convex chain to some basis. The set $F$ is also called the set of* final indices.

Note that if in the figure above we have $\{i_2, \ldots, i_{m+1}\} \cap F \neq \emptyset$, then the brick $(x_1, i_1, \overline{x_1}, i_2)$ is linked via a convex chain with a basis.

**Lemma 4.18** *Let $n, m, f \in \mathbb{N}$ and $\mathcal{B} = ((, , ^{-}), (I, \leq, \varphi_I), left, B)$ be a system of boundary equations with $|, | = 2n$. Let $F \subseteq I$ have size $f$. Suppose that every brick $b \in B$ can be linked via a convex chain of length at most $m$ to a basis with respect to $F$. Then we can bound the size of $B$ by*

$$|B| \leq (2n)^m \cdot 2f.$$

*Proof*. Consider a convex chain of length $k$, $k \leq m$, where the last brick is a basis:

$$C = ((x_1, i_1, \overline{x_1}, i_2), (x_2, i_2, \overline{x_2}, i_3), \ldots, (x_k, i_k, \overline{x_k}, i_{k+1}))$$

Given any pair $x \in , , j \in I$, there exists at most one brick $(x, i, \overline{x}, j) \in B$. Therefore the whole chain is uniquely defined by the sequence $(x_1, x_2, \ldots, x_k, i_{k+1}) \in , ^k \times F$. The number of these chains is bounded by $|, |^k \cdot |F|$. Finally, observe that every brick $b \in B$ can be linked via a convex chain either of length $m \Leftrightarrow 1$ or of length $m$ to some basis. Indeed, if $(b, b_2, \ldots, b_{k-1}, b_k)$ is a convex chain of length $k$, then $(b, b_2, \ldots, b_{k-1}, b_k, \overline{b_k}, b_k)$ is a convex chain of length $k+2$. Therefore we obtain

$$|B| \leq |, |^{m-1} \cdot |F| + |, |^m \cdot |F| \leq (2n)^m \cdot 2f.$$

□

**Remark 4.19** *Every system of boundary equations $\mathcal{B}$ satisfies the convex chain condition with respect to the set $I$, trivially. Furthermore, if we construct $\mathcal{B}$ by starting from a word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \leq g < d$, then we have $|I| \leq d$. The transformation rules below will neither increase the number $2n$ of variables nor the sum $2n + f$. It will increase the sizes of $I$ and of $B$. However, Lem. 4.18 says that a large number of boundary equations (i.e., a large set of bricks) yields that there are long convex chains in order to satisfy the convex chain condition (pictorially: many bricks build* skyscrapers*). The next step is to show that long convex chains lead to high domino towers (pictorially: skyscrapers hide high towers) and hence to a lower bound on the exponent of periodicity in any solution.*

**Proposition 4.20** *Let $n, m \in \mathbb{N}$ and $\mathcal{B} = ((, , ^{-}), (I, \leq, \varphi_I), left, B)$ be a solvable system of boundary equations with $|, | = 2n$. Let $w \models (I, \leq, \varphi_I)$ be a solution of $\mathcal{B}$. Suppose that $B$ contains a clean convex chain of length at least $m$. Then we have the following lower bound for the exponent of periodicity of the solution $w$:*

$$m \leq 4n^2 \cdot (\exp(w) + 1) \Leftrightarrow 1$$

*Proof*. The hypothesis of the proposition implies $n \neq 0$, hence $w \neq 1$. Then the assertion becomes trivial for $m < 8n^2$. Hence let $n \geq 1$ and $\lceil \frac{m+1}{4n^2} \rceil \geq 2$.

Since $w$ is a solution we may assume that $I$ is a subset of positions of $w$ and it holds that $\varphi_I(\ell, r) = \varphi(w(\ell, r))$ for all $\ell, r \in I$, $\ell \leq r$. For all $x \in ,$ define a word $w(x) \in A^*$ by

$$w(x) = w(\text{left}(x), \text{right}(x)).$$

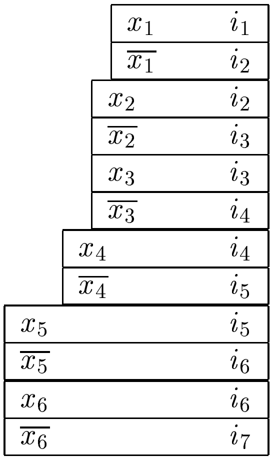
This permits also a notion of $w$-length for $x \in , .$ We define

$$|x|_w = |w(x)|.$$

Note that $w(x) = w(\overline{x})$ and hence $|x|_w = |\overline{x}|_w$ for all $x \in , .$ Let $C = (b_1, \ldots, b_m)$ be a clean convex chain of length $m$, where $b_p = (x_{i_p}, i_p, \overline{x_{i_p}}, i_{p+1})$ for all $1 \leq p \leq m$. Define $m' = \lceil \frac{m+1}{2} \rceil$, then by duality (replacing $C$ by $\overline{C}$) we may assume:

$$\text{left}(\overline{x_1}) \geq \text{left}(x_2), \quad \text{left}(\overline{x_2}) \geq \text{left}(x_3), \ldots, \quad \text{left}(\overline{x_{m'-1}}) \geq \text{left}(x_{m'}).$$

The upper part of the chain $C$ up to $m'$ might look like in in the following figure, where e.g. $m = 11$.



In the following we need a long chain where the label of the last brick has minimal $w$-length. In order to find such a chain we scan $(b_1, \ldots, b_{m'})$ from right to left. We find a sequence of indices

$$0 = p_0 < p_1 < \cdots < p_{k-1} < p_k = m'$$

such that $k \leq n$ and for all $q, j$ where $p_{j-1} < q \leq p_j$, $1 \leq j \leq k$ we have:

$$|x_q|_w \geq |x_{p_j}|_w.$$

This means that in each interval $[p_{j-1} + 1, p_j]$ the last label $x_{p_j}$ has minimal $w$-length. By the pigeon hole principle there is at least one index $j \in \{1, \ldots, k\}$ such that

$$p_j \Leftrightarrow p_{j-1} \geq \frac{m+1}{2n}.$$

We conclude that (after renaming) there is a clean convex chain $C = (b_1, \ldots, b_\ell)$ satisfying the following properties:

$$\begin{array}{rcll} \ell & = & \lceil \frac{m+1}{2n} \rceil, & \\ \text{left}(\overline{x_p}) & \geq & \text{left}(x_{p+1}) & \text{for} \quad 1 \leq p < \ell, \\ |x_p|_w & \geq & |x_\ell|_w & \text{for} \quad 1 \leq p \leq \ell. \end{array}$$

Next define $h$ (which will become the hight of a domino tower) by $h = \lceil \frac{m+1}{4n^2} \rceil$. Then it holds $h \geq 2$ and

$$2n(h \Leftrightarrow 1) + 1 \leq \left\lceil \frac{m+1}{2n} \right\rceil .$$

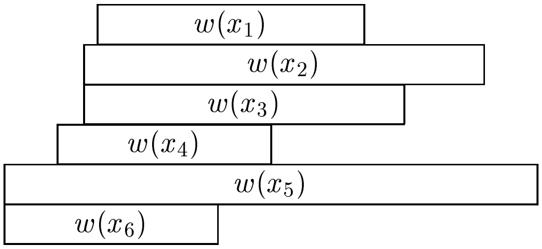
Hence there is some index $p$, $1 \leq p < \ell$ such that the label $x_p$ occurs at least $h$ times in the the clean convex chain $C$. We may assume that this is the first label $x_1$ and still $\ell \geq h$. Hence, there is a clean convex chain $C = (b_1, \ldots , b_\ell)$, which satisfies the following properties:

$$\begin{array}{rcll} \ell & \geq & h, & \\ \mathrm{left}(\overline{x_p}) & \geq & \mathrm{left}(x_{p+1}) & \text{for} \quad 1 \leq p < \ell, \\ |x_p|_w & \geq & |x_\ell|_w & \text{for} \quad 1 \leq p \leq \ell, \\ \multicolumn{4}{c}{\text{the label } x_1 \text{ occurs exactly } h \text{ times.}} \end{array}$$

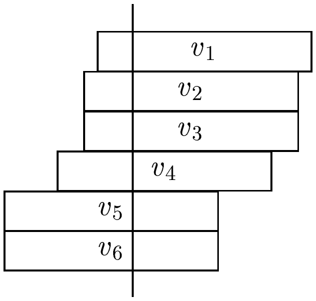
This is the point where we switch from the chain to the sequence of words:

$$(w(x_1), \ldots , w(x_\ell)).$$

We obtain a tower of words where $w(x_\ell)$ has minimal length and the word $w(x_1)$ occurs at least $h$ times.



Define $v_p \in A^*$ to be the prefix of $w(x_p)$ of length $|w(x_\ell)|$ and let $u_p = w(\mathrm{left}(x_p), i_p)$ for $1 \leq p \leq \ell$. Since $|u_p| \leq |w(\mathrm{left}(x_\ell), i_\ell)| \leq |v_\ell| = |v_p|$, the word $u_p$ is a prefix of $v_p$ for all $1 \leq p \leq \ell$. The sequence $(v_1, \ldots , v_\ell)$ can be arranged in a tower of words which is already in better shape: All words $v_p$ have equal length.



The vertical line corresponds to the factorization $v_p = u_p u'_p$ for $1 \leq p \leq \ell$.

Finally, let $\{q_1, q_2, \ldots , q_h\}$ be a set of the $h$ indices where the bricks have label $x_1$. Since the convex chain leading to this tower is clean, we see that $u_{q_j} \neq u_{q_k}$ for all $1 \leq j, k \leq h,\ j \neq k$. (This is the only point where it is used that the chain is clean!) We obtain:

$$0 \leq |u_{q_1}| < |u_{q_2}| < \cdots < |u_{q_h}|.$$

Moreover, we have $v_1 = v_{q_1} = v_{q_2} = \cdots = v_{q_h}$. We omit all other words in the tower above and we see that the word $v_1$ can be arranged in a domino tower of height $h$ and $h \geq 2$. Applying Lem. 2.5 we obtain $h \Leftrightarrow 1 \leq \exp(w_1) \leq \exp(w)$. Since $\lceil \frac{m+1}{4n^2} \rceil \leq h$, the assertion of the proposition follows. □

**Corollary 4.21** *Let* $\mathcal{B} = ((,\ ,\ ^{-}), (I, \leq, \varphi_I), \textit{left}, B)$ *denote a solvable system of boundary equations which satisfies the convex chain condition with respect to some subset* $F \subseteq I$. *Then we have*

$$|B| \leq |,\ |^{|\Gamma|^2(\exp(\mathcal{B})+1)-1} \cdot 2|F|.$$

*If moreover* $|,\ |, |F| \in \mathcal{O}(d)$, *and* $\exp(\mathcal{B}) \in 2^{\mathcal{O}(d+\log c(S))}$, *then we have*

$$|B| \in 2^{2^{\mathcal{O}(d+\log c(S))}}.$$

*Proof*. Let $2n = |,\ |$, $f = |F|$, and $m$ be the maximal length of a clean convex chain in $B$. By Lem. 4.18 and Rem 4.16 we have

$$|B| \leq (2n)^m \cdot 2f.$$

Choose a solution $w$ such that $\exp(w) \leq \exp(\mathcal{B})$. Prop. 4.20 yields a lower bound for the exponent of periodicity for all solutions. Hence:

$$m \leq 4n^2 \cdot (\exp(w) + 1) \Leftrightarrow 1.$$

Putting things together we obtain:

$$|B| \leq (2n)^{4n^2 \cdot (\exp(w)+1)-1} \cdot 2f \leq (2n)^{4n^2 \cdot (\exp(\mathcal{B})+1)-1} \cdot 2f.$$

The result follows. □

## 4.4 Transformation rules

We are ready to define the (non-deterministic) transformation rules of Makanin's algorithm. If we apply a rule to a system $\mathcal{B} = ((,\ ,\ ^{-}), (I, \leq, \varphi_I), \mathrm{left}, B)$, then the new system is denoted by $\mathcal{B}' = ((,\ ',\ ^{-}), (I', \leq, \varphi_{I'}), \mathrm{left}', B')$. The transformation rules below will have the property that if $\mathcal{B} = ((,\ ,\ ^{-}), (I, \leq, \varphi_I), \mathrm{left}, B)$ satisfies the convex chain condition with respect to some subset $F \subseteq I$, then $\mathcal{B}'$ satisfies the convex chain condition with respect to some subset $F' \subseteq I'$ such that $|,\ '| + |F'| \leq |,\ | + |F|$. Thus, if we start with a system $\mathcal{B}_0$ where $|,\ _0| = 2n_0$ and $|I_0| \leq d$, then throughout the whole procedure the size of the set of final indices is smaller than or equal to $2n_0 + d$.

We say that a (non-deterministic) rule is *downward correct*, if $w \in A^*$ is a solution of $\mathcal{B}$, then (for at least one non-deterministic choice) some suffix $w'$ of $w$ is a solution of $\mathcal{B}'$, and moreover either $|,\ '| < |,\ |$ or $|w'| < |w|$. Thus, applied to solvable systems at least one sequence of choices of downward correct rules leads to termination.

We say that a (non-deterministic) rule is *upward correct*, if $w' \in A^*$ is a solution of $\mathcal{B}'$ (and $\mathcal{B}'$ is the result of any non-deterministic choice), then there is word $w \in A^*$ which is a solution of $\mathcal{B}$.

**Rule 1** If there is some $x \in$ , with left$(x) =$ right$(x)$, then cancel both bricks

$$(x, \text{right}(x), \overline{x}, \text{right}(\overline{x})) \text{ and } (\overline{x}, \text{right}(\overline{x}), x, \text{right}(x))$$

from $B$. Cancel $x$ and $\overline{x}$ from , .

**Remark 4.22** *Obviously Rule 1 is upward and downward correct since we have $w(i,i) = 1$ for all words $w$ and all positions $i$ of $w$. Hence the set of solutions is the same. In order to preserve the convex chain condition we introduce two new final indices. Let $x \in$ , such that left$(x) =$ right$(x)$ and assume that $x, \overline{x}$ are canceled by Rule 1. Define $F' = F \cup \{$left$(x)$, left$(\overline{x})\}$. Consider a convex chain $C = (b_1, \ldots, b_m)$ where for some $1 < p \leq m$ the brick $b_p$ has the form $b_p = (x, \text{right}(x), \overline{x}, \text{right}(\overline{x}))$. Hence the brick $b_p$ is canceled. However, the brick $b_1$ is linked to $b_{p-1}$ via a convex chain and $b_{p-1}$ is now a basis since right$(x) =$ left$(x) \in F'$. Thus, if $\mathcal{B}$ satisfies the convex chain condition with respect to $F$, then the system $\mathcal{B}'$ (after an application of Rule 1) satisfies the convex chain condition with respect to $F'$. We have $|,'| + |F'| \leq |,| + |F|$.*

**Rule 2** If there exists some $x \in$ , with left$(x) =$ left$(\overline{x})$, then cancel all bricks $(x, j, \overline{x}, j)$ and $(\overline{x}, j, x, j)$ from $B$. Cancel $x$ and $\overline{x}$ from , .

**Remark 4.23** *Recall that for $(x, i, \overline{x}, j) \in B$ we have left$(x) =$ left$(\overline{x})$ if and only if $i = j$. Thus, if left$(x) =$ left$(\overline{x})$, then all bricks with label $x$ have the form $(x, j, \overline{x}, j)$. Again, Rule 2 is obviously upward and downward correct. For the convex chain condition consider a convex chain $C = (b_1, \ldots, b_m)$ where $b_p = (x, j, \overline{x}, j)$ for some $1 < p \leq m$. If we have $p < m$, then $C' = (b_1, \ldots, b_{p-1}, b_{p+1}, \ldots, b_m)$ is a shorter convex chain linking $b_1$ with a basis. For $p = m$ we have $j \in F$. Hence $b_{m-1}$ is also a basis.*

**Rule 3** Let $\ell = \min(I)$. If $\ell \notin$ left(, ), then cancel the index $\ell$ from $I$. This means we replace the linear order over $S$ by the induced sub-order $(I', \leq, \varphi_{I'})$ where $I' = I \setminus \{\ell\}$.

**Remark 4.24** *Clearly, the convex chain condition is not affected by this rule. Downward correctness is obvious, too. To see the upward correctness let $(I, \leq, \varphi_I)$ be given by the $S$-sequence $(s_1, \ldots, s_m)$ and let $w' \in A^*$ be a solution of the new system after an application of Rule 3 such that $\min(I')$ is the first position of $w'$. By definition of an $S$-sequence there is a non-empty word $u \in A^+$ with $\varphi(u) = s_1$. Then the first position of $w'$ is not equal to the first position in the word $uw'$, and $uw'$ is a solution of $\mathcal{B}$. For later use notice that we can choose $u$ such that $|u| \leq |S|$.*

The next rule is very complex. It is the heart of the algorithm. Before we apply it to some system $\mathcal{B} = ((,\ ,^{-}), (I, \leq, \varphi_I), \text{left}, B)$, we apply Rules 1, 2 or 3 as often as possible. In particular, we shall assume that left$(x) <$ right$(x)$, left$(x) \neq$ left$(\overline{x})$ for all $x \in$ , , and that there exists some $x \in$ , with left$(x) = \min(I)$.

**Rule 4** We divide Rule 4 into six steps.

We need some notation. Define $\ell = \min(I)$ and $r = \max\{\text{right}(x) \mid x \in\ ,\ , \text{left}(x) = \ell\}$. Note that $\ell \in$ left(, ), hence $r \in I$ exists and we have $\ell < r$. Choose (and fix) some $x_o \in$ , with $\text{left}(x_o) = \ell$ and $\text{right}(x_o) = r$. Define $\ell^* = \text{left}(\overline{x_o})$ and $r^* = \text{right}(\overline{x_o})$. Define the *critical boundary* $c \in I$ by $c = \min\{c', r\}$ where

$$c' = \min\{\text{left}(x) \mid x \in , , \; r < \text{right}(x)\}.$$

Note that since $r < r^* = \text{right}(\overline{x_o})$, the minimum $c'$ and hence the critical boundary $c$ exists. We have $\ell < c \leq r < r^*$ and $c \leq \ell^* < r^*$, but the ordering of $r$ and $\ell^*$ depends on the system.

Define the subset $T \subseteq I$ of *transport positions* by

$$T = \{i \in I \mid i \leq c\} \cup \{i \in I \mid \exists (x, i, \overline{x}, j) \in B : \text{left}(x) < c\}$$

Note that $\min(T) = \ell$ and that $i \in T$ for all $(x_o, i, \overline{x_o}, j) \in B$. Moreover, since $\text{left}(x) < c$ implies $\text{right}(x) \leq r$, we have $\max(T) = r$.

**Step 1** Choose some admissible extension $(I^*, \leq, \varphi_{I^*})$ of $(I, \leq, \varphi_I)$ by $T$ at $[\ell^*, r^*]$. By convention we identify $I$ as a subset of $I^*$, whence $I \subseteq I^*$, and there is a subset $T^* \subseteq I^*$ with $\min(T^*) = \ell^*$, $\max(T^*) = r^*$, and which is in order respecting bijection with $T$. For each $i \in T$ the corresponding position in $T^*$ is denoted $i^*$. Having these notations we put a further restriction on the admissible extension: We consider only those admissible extensions where first, $i < i^*$ for all $i \in T$ and second:

$$\begin{aligned} \text{left}(x)^* \leq \text{left}(\overline{x}) \quad &\Leftrightarrow \quad i^* \leq j, \\ \text{left}(x)^* \geq \text{left}(\overline{x}) \quad &\Leftrightarrow \quad i^* \geq j \end{aligned}$$

for all $(x, i, \overline{x}, j) \in B$ with $\text{left}(x) < c$. Note that for all $(x_o, i, \overline{x_o}, j) \in B$ this implies $i^* = j$. If such an admissible extension is not possible, then Step 1 cannot be completed and Rule 4 is not applicable.

**Step 2** Introduce new variables $x_\nu$ and $\overline{x_\nu}$ and define $\text{left}(x_\nu) = c$, $\text{left}(\overline{x_\nu}) = c^*$. For all $i \in T$ such that there is some $(x, i, \overline{x}, j) \in B$ with $\text{left}(x) < c \leq i$ introduce new bricks $(x_\nu, i, \overline{x_\nu}, i^*)$ and $(\overline{x_\nu}, i^*, x_\nu, i)$.

**Step 3** As long as there is a variable $x \in ,$ with $\text{left}(x) < c$, replace $\text{left}(x)$ by $\text{left}'(x) = \text{left}(x)^*$ and replace all bricks $(x, i, \overline{x}, j), (\overline{x}, j, x, i) \in B$ by $(x, i^*, \overline{x}, j)$ and $(\overline{x}, j, x, i^*)$.

**Remark 4.25** *To have some notation let $x$ denote a variable before Step 3 and let $x'$ be the corresponding variable after Step 3. Likewise let $b = (x, i, \overline{x}, j)$ denote a brick before Step 3 and let $b' = (x', i', \overline{x}', j)$ be the corresponding brick after Step 3. If left$(x)$ = left$'(x')$, then sometimes we may still write $x = x'$. In particular, $x_\nu = x'_\nu$, $\overline{x_\nu} = \overline{x_\nu}'$, $\overline{x_o} = \overline{x_o}'$, but $x_o \neq {x_o}'$.*

*For $b = (x, i, \overline{x}, j)$ and $b' = (x', i', \overline{x}', j')$ there are four cases:*

$b' = (x', i^*, \overline{x}', j^*)$ *if left$(x) < c$, left$(\overline{x}) < c$,*
$b' = (x', i^*, \overline{x}, j)$ *if left$(x) < c$, $c \leq$ left$(\overline{x})$,*
$b' = (x, i, \overline{x}', j^*)$ *if $c \leq$ left$(x)$, left$(\overline{x}) < c$,*
$b' = (x, i, \overline{x}, j)$ *if $c \leq$ left$(x)$, $c \leq$ left$(\overline{x})$.*

*Note that after Step 3 all bricks $(x_o, i, \overline{x_o}, j) \in B$ have the form $(x'_o, i^*, \overline{x_o}, i^*)$.*

**Step 4** Define as the new set of final indices

$$F' = \{i^* \in I^* \mid i < c \text{ and } i \in F\} \cup \{i \in F \mid c \leq i\}.$$

**Step 5** Cancel all bricks with label $x'_o$ or $\overline{x_o}$, i.e., cancel all bricks of the form $(x'_o, i^*, \overline{x_o}, i^*)$ or $(\overline{x_o}', i^*, x'_o, i^*)$. Then cancel the variables $x_o$, $\overline{x_o}$.

**Step 6** Replace $I^*$ by $I' = \{i \in I^* \mid c \leq i\}$ and consider the linear order $(I', \leq, \varphi_{I'})$ induced by $I' \subseteq I^*$.

After Step 6 the transformation rule is finished. The new system is denoted by $\mathcal{B}' = ((,', ^-), (I', \leq, \varphi_{I'}), \text{left}', B')$. We will show from Lem. 4.30 to 4.33 below that $\mathcal{B}'$ satisfies the convex chain condition with respect to $F'$. The first lemma is a trivial observation.

**Lemma 4.26** *We have* $|,'| = |,|$ *and* $|F'| \leq |F|$.

*Proof*. In Step 2 new variables $x_\nu$ and $\overline{x_\nu}$ are introduced, but in Step 5 the variables $x'_0$ and $\overline{x_0}$ are canceled. Hence $|,'| = |,|$. The set of final indices is changed in Step 4. However, the assertion $|F'| \leq |F|$ is clear by the definition of $F'$. □

The following lemma is crucial to bound the size of $I$ during the transformation procedure. The lemma has a rather subtle proof.

**Lemma 4.27** *Let* $\beta' = |\{(x', i', \overline{x}', j') \in B' \mid \mathit{left}'(x') < i'\}|$ *and* $\beta = |\{(x, i, \overline{x}, j) \in B \mid \mathit{left}(x) < i\}|$. *Then we have*

$$2|I'| \Leftrightarrow \beta' \leq 2|I| \Leftrightarrow \beta.$$

*Proof*. The inequality can be destroyed either by a new position $i^* \in T^* \setminus I$ or by the cancelation of bricks $(x'_o, i^*, \overline{x_o}, i^*)$, $(\overline{x_o}, i^*, x'_o, i^*)$ in Step 5, where $\ell^* < i^*$. (Recall the definition of $\beta$ and $\beta'$ and that $\text{left}(x_o) = \ell$, $\text{left}'(x'_o) = \ell^*$.) The cancelation of these bricks involves again a position of type $i^* \in T^*$. Fortunately, if $(x'_o, i^*, \overline{x_o}, i^*)$ is canceled, where $\ell^* < i^*$, then $i^* = j$ for some $j \in I \setminus \{\ell\}$. In particular, $i^*$ is not a new position and the two cases don't occur simultaneously. Therefore it is enough to find for each $i^* \in T^* \setminus \{\ell^*\}$ either two new bricks which are introduced in Step 2 or one position which is canceled in Step 6. Then the total balance will be negative or zero.

Let us consider the positions of type $i^* \in T^* \setminus \{\ell^*\}$ one by one. If $c^* < i^*$, then by the definition of $T$ and Step 2 there are two new bricks $(x_\nu, i, \overline{x_\nu}, i^*)$, $(\overline{x_\nu}, i^*, x_\nu, i) \in B'$ and we have $\text{left}(x_\nu) < i$, $\text{left}(\overline{x_\nu}) < i^*$. Next consider $i^* = c^*$. At least one position (namely $\ell$) is canceled in Step 6. Next let $\ell^* < i^* < c^*$, i.e., $\ell < i < c$. The position $i$ is canceled in Step 6. Hence we have the assertion of the lemma. □

**Lemma 4.28** *Rule 4 is downward correct.*

*Proof*. Let $w \in A^*$ be a solution of $\mathcal{B}$. Since $w \models (I, \leq, \varphi_I)$, we can view $I$ as a subset of positions of $w$ with $\ell = 0$. Let $w = vw'$ where $v = w(\ell, c)$. The word $v$ is a non-empty prefix of $w(\ell, r)$. The word $w(\ell, r)$ is a prefix of $w$ and at the same time another factor of $w'$; we have $w(\ell, r) = w(\ell^*, r^*)$ with $\ell < \ell^*$ due to the brick $(x_o, r, \overline{x_o}, r^*) \in B$. The set $T$ is a subset of positions of $w(\ell, r)$, hence we find a corresponding subset $T^*$ of positions of $w(\ell^*, r^*)$. The union $I \cup T^*$ leads to an admissible extension $(I^*, \leq, \varphi_I)$ such that first, $i < i^*$ for all $i \in T$ and second, $w(j, k) = w(j^*, k^*)$ for all $j, k \in T, j \leq k$. A careful but easy inspection of Rule 4 then shows that $w' \models (I', \leq, \varphi_{I'})$ and $w'$ is a solution of $\mathcal{B}'$. □

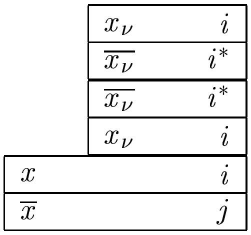
**Lemma 4.29** *Rule 4 is upward correct.*

*Proof*. Let $w' \in A^*$ be a solution of $\mathcal{B}'$. Since $w' \models (I', \leq, \varphi'_I)$, we can view $I'$ as a subset of positions of $w'$ where $c$ is the first position of $w'$. Define $v = w'(l^*, c^*)$ and let $w = vw'$. Then we have $w \models (I^*, \leq, \varphi_{I^*})$ such that $v = w(l, c) = w(l^*, c^*)$. With the help of the bricks $(x_\nu, i, \overline{x_\nu}, i^*)$ we conclude that $w(j, k) = w(j^*, k^*)$ for all $j, k \in T, j \leq k$. Therefore we have $w(\text{left}(x), i) = w(\text{left}(\overline{x}), j)$ for all $(x, i, \overline{x}, j) \in B$. Since $I \subseteq I^*$, we have $w \models (I, \leq, \varphi_I)$ and $w$ is a solution of $\mathcal{B}$. □

Finally we show that Rule 4 preserves the convex condition. This is clear for Step 1, for the other steps we state lemmata.

**Lemma 4.30** *Step 2 preserves the convex chain condition with respect to the set $F$.*

*Proof*. The new bricks in Step 2 have the form $(x_\nu, i, \overline{x_\nu}, i^*)$ and $(\overline{x_\nu}, i^*, x_\nu, i)$ for some $(x, i, \overline{x}, j) \in B$ with $\text{left}(x) < c = \text{left}(x_\nu) \leq i$. Since $(x, i, \overline{x}, j) \in B$ can be linked via a convex chain to some basis, it is enough to consider the following figure:



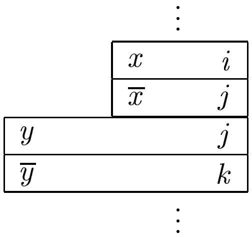
□

**Lemma 4.31** *Let $C = (b_1, \ldots, b_m)$ be a convex chain before Step 3 linking $b_1$ with $b_m$. Then after Step 3 there is a convex chain $C'$ linking $b'_1$ with $b'_m$.*

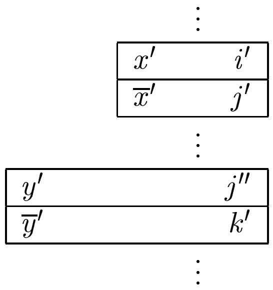
*Proof*. Let us have a local look at the convex chain:

$$C = (\ \ldots, (x, i, \overline{x}, j), (y, j, \overline{y}, k) \ldots\ ).$$

By symmetry we may assume that $\text{left}(\overline{x}) \geq \text{left}(y)$. Pictorially this local part is then given by the following figure.



This is the situation before Step 3. After Step 3 let us denote the corresponding bricks by $(x', i', \overline{x}', j')$ and $(y', j'', \overline{y}', k')$. This yields the following figure.



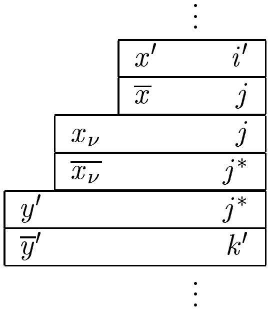
The question is whether or not $j' = j''$. If $j' = j^*$ or $j'' = j$, then we have $j' = j''$, and the chain is not broken. Hence we have to consider the case $j' = j$ and $j'' = j^*$, only. This case is equivalent to

$$\text{left}(y) < c \leq \text{left}(\overline{x}) \leq j.$$

With the help of the brick $(x_\nu, j, \overline{x_\nu}, j^*)$, which was introduced in Step 2, we can repair the broken chain. We have

$$\text{left}(x_\nu) = c \leq \text{left}(\overline{x}), \quad \text{left}'(y') < c^* = \text{left}(\overline{x_\nu})$$

and we obtain the following figure:



Doing this transformation wherever necessary we construct the convex chain $C'$. □

Note that $C'$ constructed in the lemma above may contain many bricks of the form $(x'_o, i^*, \overline{x_o}, i^*)$ and $(\overline{x_o}, i^*, x'_o, i^*)$. These bricks were canceled only later in Step 5. In fact their presence in the next lemma is very useful again.

**Lemma 4.32** *After Step 4 the convex chain condition is satisfied with respect to the set $F'$.*

*Proof*. Let $b'$ be a brick after Step 3 and $b$ the corresponding brick before Step 3. This brick $b$ is linked before Step 3 via a convex chain to some basis $(x, i, \overline{x}, j)$ with $j \in F$. Lem. 4.31 states that after Step 3 the brick $b'$ is linked via a convex chain to the corresponding brick $(x', i', \overline{x}', j')$. For $j < c$ we have $\text{left}(\overline{x}) < c$ and $j' = j^* \in F'$. Hence $(x', i', \overline{x}', j^*)$ is again a basis. For $j' = j$ we have $c \leq j$ and therefore $j \in F'$. This solves also the case $j' = j$. The remaining case is $c \leq j$ and $j' = j^*$. This means $\text{left}(\overline{x}) < c \leq j$. By Step 2 there is a brick $(\overline{x_\nu}, j^*, x_\nu, j)$ and we have $\text{left}'(\overline{x}') < c^* = \text{left}(\overline{x_\nu})$. We may put the brick $(x', i', \overline{x}', j^*)$ upon the basis $(\overline{x_\nu}, j^*, x_\nu, j)$. Since $j \in F \cap F'$, it is in fact a basis before and after Step 4. We obtain the following figure:

| | | |
|---|---|---|
| $x'$ | | $i'$ |
| $\overline{x}'$ | | $j^*$ |
| | $\overline{x_\nu}$ | $j^*$ |
| | $x_\nu$ | $j$ |

□

**Lemma 4.33** *Steps 5 and 6 preserve the convex chain condition with respect to the set $F'$.*

*Proof*. Step 5 is a special case of an application of Rule 2, likewise Step 6 is a special case of applications of Rule 3. In particular, the convex chain condition is preserved. □

The lemmata above yield to the following proposition:

**Proposition 4.34** *Rule 4 is upward and downward correct. It preserves the convex chain condition.*

**Example 4.35** *Let $x_1 \cdots x_g = x_{g+1} \cdots x_d$ be a word equation, $1 \leq g < d$ such that the regular constraints are given by a mapping $\psi : \Omega \to S$. Let*

$$\mathcal{B} = ((,\ ,^{-}), (I, \leq, \varphi_I), \mathit{left}, B)$$

*be the result of the (logspace-) reduction presented in Sect. 4. Recall that $(I, \leq, \varphi_I)$ represents the $S$-sequence*

$$(\psi(x_1), \ldots, \psi(x_g), \psi(x_{g+1}), \ldots, \psi(x_d)).$$

*We may assume that $(I, \leq, \varphi_I)$ is in its standard representation, $I = \{0, \ldots, d\}$. According to the reduction the set , contains two variables $x_0$ and $\overline{x_0}$ such that $\mathit{left}(x_0) = 0$, $\mathit{right}(x_0) = g = \mathit{left}(\overline{x_0})$, and $\mathit{right}(\overline{x_0}) = d$. The set $B$ contains at most $d$ boundary equations (or bricks), among them there is the brick:*

| | |
|---|---|
| $x_0$ | $g$ |
| $\overline{x_0}$ | $d$ |

*We have $|I| = d + 1$ and $|,| = |B| \leq 2d$. If the word equation has a non-singular solution satisfying the regular constraints, then $\exp(\mathcal{B}) \leq 2 \cdot e(c(S), d)$.*

*Rules 1 to 3 are not applicable to $\mathcal{B}$, but we can try Rule 4. Doing this we find:*

$$x_o = x_0, \quad l = 0, \quad c = g = r = l^*, \quad \text{and } c^* = g^* = r^* = d.$$

*The set $T$ of transport positions is $T = \{0, \ldots, g\}$.*

*In Step 1 we have to choose some admissible extension of $(I, \leq, \varphi_I)$ by $T$ at $[g, d]$. In general it is not clear that such an extension exists. Under the hypothesis that $x_1 \cdots x_g = x_{g+1} \cdots x_d$ has a non-singular solution $\sigma : \Omega \to A^+$ with $\varphi \circ \sigma = \psi$ we can continue. Let $v = \sigma(x_1 \cdots x_g)$ and assume that $v$ has minimal length among all solutions satisfying the regular constraints given by $\psi$. With the help of this word Step 1 can be completed: Define $w = vv$, then we have*

$$w \models (\psi(x_1), \ldots, \psi(x_d)).$$

*The set of positions of $w$ is $\{0,\ldots,m,m+1,\ldots,2m\}$ where $m=|v|$. The fact that $w$ is a model of $(I,\leq,\varphi_I)$ is realized by an order respecting injective mapping*

$$\rho:\{0,\ldots,d\}\to\{0,\ldots,2m\}.$$

*Define $T^*=\{m+\rho(i)\mid 0\leq i\leq g\}$ and $I^*=\rho(I)\cup T^*$. Since $I^*$ is a subset of positions of $w$, this induces a linear suborder over $S$, which is denoted by $(I^*,\leq,\varphi_{I^*})$. We have $|I^*|\leq d+g\Leftrightarrow 1$. After renaming we may assume $I^*=\{0,\ldots,d\}\cup T^*$, $T^*=\{0^*,\ldots,g^*\}$ such that $0^*=c=g$, $c^*=g^*=d$. This completes Step 1 of Rule 4. Since in reality we usually do not know $v$, the choice of $I^*$ is a non-deterministic guess!*

*The next steps in Rule 4 are deterministic. In Step 2 we introduce new variables $x_\nu$ and $\overline{x_\nu}$ with left$(x_\nu)=g=$ right$(x_\nu)$ and left$(\overline{x_\nu})=d=$ right$(\overline{x_\nu})$.*

*In Step 3 we transport the structure of the interval $[0,g]$ to $[0^*,g^*]=[g,d]$. If we still view $I^*$ as a subset of positions of $w$, then this reflects a transport to the positions from the first to the second factor $v$ in the word $w=vv$.*

*The definition of $F'$ according to Step 4 is*

$$F'=\{i\in I^*\mid g\leq i\}.$$

*In Step 5 we cancel the bricks $(x_o,d,\overline{x_o},d)$, $(\overline{x_o},d,x_o,d)$ and the variables $x_o$, $\overline{x_o}$.*

*In Step 6 we replace $I^*$ by $I'=F'$.*

*Rule 4 is finished. The cardinality of $I'$ is bounded by $d$. Let $\mathcal{B}'$ denote the new system, then the word $v$ is a solution, $v\models(I',\leq,\varphi_(I')$.*

*Since in the present situation left$(x_\nu)=$ right$(x_\nu)=g$, Rule 1 is now applicable to $\mathcal{B}'$, it cancels the superfluous bricks $(x_\nu,g,\overline{x_\nu},d)$, $(\overline{x_\nu},d,x_\nu,g)$ and the variables $x_\nu$ and $\overline{x_\nu}$. The new system after an application of Rule 1 is denoted by $\mathcal{B}''=((,{}''_0,{}^-),(I''_0,\leq,\varphi_{I''_0}),$ left$''_0,B''_0)$. We have $|I''|\leq d$, $|,''|=|B''|\leq 2(d\Leftrightarrow 1)$. It is now the word $v$ which is a solution of $\mathcal{B}''$, hence $\exp(\mathcal{B}'')\leq\exp(v)$. Therefore, we can choose $e(\mathcal{B}'')=e(c(S),d)$.*

# 5 Proof of Theorem 4.12

## 5.1 Decidability

The proof of Thm. 4.12 is a reduction to a reachability problem in some finite directed graph. The implications for space- and time bounds for Makanin's algorithm are given later.

The instance is a system of boundary equations

$$\mathcal{B}_0=((,{}_0,{}^-),(I_0,\leq,\varphi_{I_0}),\mathrm{left}_0,B_0).$$

We may assume that $\mathcal{B}_0$ satisfies the assumptions made at the beginning of Sect. 4.3, because otherwise $\mathcal{B}_0$ is not solvable. For trivial reasons the system $\mathcal{B}_0$ satisfies the convex chain condition with respect to the set $F_0=I_0$.

Let $2n_0=|,{}_0|$ and $f_0=|F_0|=|I_0|$. According to Prop. 4.13 choose a number $e(\mathcal{B}_0)$ such that either $\mathcal{B}_0$ is not solvable or $\exp(w)\leq e(\mathcal{B}_0)$ for some solution $w$ of $\mathcal{B}_0$. Define an integer $\beta_{\max}$ by

$$\beta_{\max}=(2n_0)^{4n_0^2(e(\mathcal{B}_0)+1)-1}\cdot 2(2n_0+f_0).$$

Note that this value is defined just to fit Cor. 4.21 for a set of final indices having size at most $2n_0+f_0$.

Now, define a directed graph $\mathcal{G}$ (the search graph of Makanin's algorithm) as follows. The nodes of the search graph $\mathcal{G}$ are the systems of boundary equations $\mathcal{B} = ((,\ ,^{-}), (I, \leq, \varphi_I), \mathrm{left}, B)$, where:

$$\begin{aligned} |,\,| &\leq 2n_0,\\ |I| &\leq \frac{n_0+2}{2}\cdot\beta_{\max},\\ |B| &\leq \beta_{\max}. \end{aligned}$$

For systems $\mathcal{B}$,$\mathcal{B}' \in \mathcal{G}$ we define an arc from $\mathcal{B}$ to $\mathcal{B}'$ whenever first, there is a transformation rule is applicable to $\mathcal{B}$ and second, $\mathcal{B}'$ is the result of the corresponding transformation. A system $\mathcal{B} \in \mathcal{G}$ with an empty set of variables is called a *terminal node.*

Clearly, $\mathcal{B}_0 \in \mathcal{G}$ and the search graph $\mathcal{G}$ has only finitely many nodes. Hence, it is enough to show the following claim: The system $\mathcal{B}_0$ has a solution if and only if there is a directed path in $\mathcal{G}$ from $\mathcal{B}_0$ to some terminal node.

The "if"-direction of the claim is trivial since all transformation rules are upward correct and since all terminal nodes are solvable by Lem. 4.6. For the "only-if"-direction let $\mathcal{B}_0$ be solvable and let $w_0 \models (I_0, \leq, \varphi_{I_0})$ be a solution satisfying $\exp(w_0) \leq \exp(\mathcal{B}_0)$.

Let $M \geq 0$ and assume that there is an inductively defined sequence of solvable systems $(\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_M)$, $M \geq 0$ such that the following properties are satisfied for all $1 \leq k \leq M$:

- $\mathcal{B}_k = ((,\ {}_k, ^{-}), (I_k, \leq, \varphi_{I_k}), \mathrm{left}_k, B_k)$ is the result of some transformation rule applied to $\mathcal{B}_{k-1}$,
- $\mathcal{B}_k$ has a solution $w_k \models (I_k, \leq, \varphi_{I_k})$ such that $w_k$ is a suffix of $w_{k-1}$,
- either $|,\ {}_k| < |,\ {}_{k-1}|$ or $|w_{k-1}| < |w_k|$,
- $\mathcal{B}_k$ satisfies the convex chain condition with respect to some subset $F_k \subseteq I_k$ with $|F_k| \leq 2n_0 \Leftrightarrow |,\ {}_k| + f_0$.

If $\mathcal{B}_M$ is a system of boundary equations without variables, then we stop. Otherwise, since $\mathcal{B}_M$ is solvable, a transformation rule is applicable. Consequently, the sequence can be continued by some solvable system $\mathcal{B}_{M+1}$ satisfying all properties above. The third property however implies that $M \leq n_0 + |w_0|$. Hence, finally we must reach a system without variables. We may assume that this happens with reaching $\mathcal{B}_M$. Let us show that all $\mathcal{B}_k$ are nodes of $\mathcal{G}$ for all $0 \leq k \leq M$. This will imply the claim since then there is a directed path to $\mathcal{B}_M$, and $\mathcal{B}_M$ is a terminal node.

We have to verify $|,\ {}_k| \leq 2n_0$, $|I_k| \leq \frac{n_0+2}{2}\cdot\beta_{\max}$, and $|B_k| \leq \beta_{\max}$.

The assertion $|,\ {}_k| \leq 2n_0$ is trivial. The second property of the sequence implies $\exp(\mathcal{B}_k) \leq \exp(w_k) \leq \exp(w_0) \leq e(\mathcal{B}_0)$. By Cor. 4.21 and the fourth property we have $|B_k| \leq \beta_{\max}$. The next lemma yields an invariant which will give the desired bound on the size of every $I_k$.

**Lemma 5.1** *For* $0 \leq k \leq M$ *define* $\beta_k = |\{(x, i, \overline{x}, j) \in B_k \mid \mathit{left}_k(x) < i\}|$. *Then for all* $1 \leq k \leq M$ *we have:*

$$2\,|I_k| \Leftrightarrow \beta_k + \frac{|,\ {}_k|}{2}\cdot\beta_{\max} \leq 2\,|I_{k-1}| \Leftrightarrow \beta_{k-1} + \frac{|,\ {}_{k-1}|}{2}\cdot\beta_{\max}.$$

*Proof*. Consider the rule which was applied to pass from $\mathcal{B}_{k-1}$ to $\mathcal{B}_k$. For Rule 1 or 2 we have:

$$\begin{aligned}
|,_k| &= |,_{k-1}| \Leftrightarrow 2, \\
|I_k| &= |I_{k-1}|, \\
\beta_{k-1} \Leftrightarrow \beta_k &\leq \beta_{\max}.
\end{aligned}$$

For Rule 3 we have:

$$\begin{aligned}
|,_k| &= |,_{k-1}|, \\
|I_k| &= |I_{k-1}| \Leftrightarrow 1, \\
|\beta_k| &= |\beta_{k-1}|.
\end{aligned}$$

Finally, for Rule 4 we have $|,_k| = |,_{k-1}|$ and Lem. 4.27 says:

$$2\,|I_k| \Leftrightarrow \beta_k \leq 2\,|I_{k-1}| \Leftrightarrow \beta_{k-1}.$$

The assertion of the lemma follows. □

A consequence of Lem. 5.1 (and $\beta_k \leq \beta_{\max}$) is:

$$2\,|I_k| \leq 2\,|I_0| + (n_0 + 1)\beta_{\max} \text{ for all } 0 \leq k \leq M.$$

Since $|I_0| \leq \frac{1}{2}\beta_{\max}$, we obtain $|I_k| \leq \frac{n_0+2}{2}\beta_{\max}$. Hence $\mathcal{B}_k \in \mathcal{G}$ for all $0 \leq k \leq M$. This proves Thm. 4.12, hence Makanin's result.

### 5.2 Complexity in terms of the semigroup $S$ and the maximal number of boundary equations

Our estimations on the upper bounds of Makanin's algorithm are given by the size of the semigroup $S$ and the number $\beta_{\max}$ as defined in the precedent section, which is the maximal number of boundary equations.

A node $\mathcal{B} = ((,\ ,^{-}), (I, \leq, \varphi_I), \text{left}, B)$ of the search graph $\mathcal{G}$ is encoded as a binary string over $\{0, 1\}$ as follows: The code for $(,\ ,^{-})$ is simply the number $n$ written in binary such that $|,\ | = 2n$. Thus, $\mathcal{O}(\log n_0)$ bits are enough for this part. The linear order $(I, \leq, \varphi_I)$ is encoded by its underlying $S$-sequence. For this part $\mathcal{O}(n_0\beta_{\max} \log |S|)$ bits are used. The mapping $\text{left} : ,\ \to I$ is encoded by using $\mathcal{O}(n_0 \log(n_0\beta_{\max}))$ bits. Finally, the set of bricks $B$ can be encoded by using $\mathcal{O}(\beta_{\max} \log(n_0\beta_{\max}))$ bits. Note that $n_0 \leq \log \beta_{\max}$. It follows that there is effectively a constant $c \in \mathbb{N}$ such that every $\mathcal{B} \in \mathcal{G}$ can be described by a bit string of length equal to $c \cdot (\log |S| \cdot \beta_{\max} \cdot \log(\beta_{\max}))$. Up to some calculations performed over $S$ this is the essential upper space bound for the non-deterministic procedure. It is at most double exponential in the input size, but we will come back to this point later.

The number of bits we need for the code yields an upper bound for the size of $\mathcal{G}$. Using the constant $c$ above, define a natural number $\gamma_{\max}$ by:

$$\gamma_{\max} = 2^{c \cdot (\log|S| \cdot \beta_{\max} \cdot \log(\beta_{\max}))} \in 2^{\mathcal{O}(\log|S| \cdot \beta_{\max} \cdot \log(\beta_{\max}))}.$$

**Lemma 5.2** *The number of nodes in $\mathcal{G}$ is less than or equal to $\gamma_{\max}$.*

*Proof.* The number of nodes is as most exponential in the number of bits used in a description for a node. □

The following assertion is now clear:

**Proposition 5.3** *The system $\mathcal{B}_0$ is solvable if and only if $\mathcal{G}$ contains a directed path*

$$(\mathcal{B}_0, \ldots, \mathcal{B}_m)$$

*to the terminal node $\mathcal{B}_m = (\emptyset, \emptyset, \emptyset, \emptyset)$ such that $m \leq \gamma_{\max}$.*

*Proof.* If there is a path to some terminal node then this path can be elongated by applications of Rule 3 until finally the underlying linear order is the empty set. We may assume that this path is without cycles, then Lem. 5.2 implies $m \leq \gamma_{\max}$. □

**Corollary 5.4** *Let $\mathcal{B}_0$ be solvable and let $w_0 \models (I_0, \leq, \varphi_{I_0})$ be a solution, where the length $|w_0|$ is minimal. Then we have*

$$|w_0| \leq |S| \cdot 2^{\gamma_{\max}}.$$

*Proof.* Consider a sequence $(\mathcal{B}_0, \ldots, \mathcal{B}_m)$, $m \leq \gamma_{\max}$ to the terminal node $\mathcal{B}_m = (\emptyset, \emptyset, \emptyset, \emptyset)$ as in Prop. 5.3. Going the path backwards we define inductively solutions $v_m, v_{m-1}, \ldots, v_0$ of the systems $\mathcal{B}_m, \mathcal{B}_{m-1}, \ldots, \mathcal{B}_0$ as follows. The initial solution is $v_m = 1$. Assume that $v_m, \ldots, v_k$, $1 \leq k \leq m$ are already defined. Depending on the transformation rule which link $\mathcal{B}_{k-1}$ to $\mathcal{B}_k$ we define the solution $v_{k-1} \models \left(I_{k-1}, \leq, \varphi_{I_{k-1}}\right)$ of the system $\mathcal{B}_{k-1}$.

For Rule 1 or 2 we define $v_{k-1} = v_k$. For Rule 3 we define $v_{k-1} = uv_k$ for some suitable $u \in A^+$. It is clear that we can choose $u$ such that $|u| \leq |S|$. Hence $|v_{k-1}| \leq |S| + |v_k|$.

For Rule 4 we find a solution $v_{k-1} = uv_k$, where $u$ is a factor of $v_k$. Hence $|v_{k-1}| \leq 2\,|v_k|$.

We end up with a solution $v_0 \models (I_0, \leq, \varphi_{I_0})$ of $\mathcal{B}_0$ such that $|v_0| \leq |S| \cdot 2^{\gamma_{\max}}$.

Since $w_0$ is of minimal length we have $|w_0| \leq |v_0|$. □

**Remark 5.5** *Assume that $\mathcal{B}_0$ is solvable and $w_0$ is a solution of minimal length. The word $w_0$ can be used as a* model *for constructing a path*

$$(\mathcal{B}_0, \ldots, \mathcal{B}_M)\,,\ M \geq 1$$

*to some terminal node. However, we cannot exclude that this path has many cycles. In particular, $M$ need not to be the number $m$, which was used in the proof of Cor. 5.4. Due to the construction of the solution $v_0$ above, it is possible that $v_0 \neq w_0$ and $e(\mathcal{B}_0) < \exp(v_0)$.*

**Corollary 5.6** *In deterministic time*

$$2^{\mathcal{O}(|S| \cdot 2^{\gamma_{\max}})}$$

*we find by exhaustive search either a solution of minimal length or we can report that $\mathcal{B}_0$ is not solvable. This upper bound is five times exponential in the input size (number of bits used in the encoding) of $\mathcal{B}_0$.*

*Proof.* Test all strings up to the length of $|S| \cdot 2^{\gamma_{\max}}$ whether they are a solution. Stop when the first solution is encountered. If no solution up to this length is found, then $\mathcal{B}_0$ is not solvable. □

## 5.3 An upper bound for the complexity of solving word equations

The original question of the chapter is whether a given word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \leq g < d$ with regular constraints has a solution. We may assume that each regular language $L_x \subseteq A^*$ is specified by an NFA with $r_x$ states, $x \in \Omega$. Define $r = \sum_{x \in \Omega} r_x$; we are going to measure the complexity of Makanin's algorithm in terms of $d$ and $r$. First, we choose a suitable semigroup $S$ and a homomorphism $\varphi : A^+ \to S$. By Rem. 3.1 we may assume that $S$ satisfies $|S| \leq 2^{r^2}$ and $c(S) \leq r!$. By Thm. 3.2 choose a value $e\left(c\left(S\right), d\right) \in c(S) \cdot 2^{\mathcal{O}(d)} \subseteq 2^{\mathcal{O}(d + r \log r)}$ such that $e\left(c\left(S\right), d\right)$ is an upper bound for the exponent of periodicity. Transform the word equation (by a non-deterministic guess) into a system of boundary equations

$$\mathcal{B}_0 = ((, {}_0, {}^-), (I_0, \leq, \varphi_{I_0}), \mathrm{left}_0, B_0).$$

such that the word equation has a solution satisfying the regular constraints if and only if $\mathcal{B}_0$ is solvable. This is possible such that first, $|I_0|, |, {}_0|, |B_0| \in \mathcal{O}(d)$, and second, if $\mathcal{B}_0$ is solvable, then

$$e\left(\mathcal{B}_0\right) \leq 2 \cdot e\left(c\left(S\right), d\right) \in 2^{\mathcal{O}(d + r \log r)}.$$

More precisely, by Ex. 4.35 we can say $|I_0| \leq d \Leftrightarrow 1$, $|, {}_0| = |B_0| \leq 2(d \Leftrightarrow 1)$ and, if $\mathcal{B}_0$ is solvable, then $e\left(\mathcal{B}_0\right) \leq e\left(c\left(S\right), d\right)$.

Compute a value $\beta_{\max} \in 2^{2^{\theta(d + r \log r)}}$ such that the search graph $\mathcal{G}$ satisfies Prop. 5.3 for the corresponding value $\gamma_{\max}$. Recall that $\beta_{\max}$ is an upper bound for the number of boundary equations of each node and that $\gamma_{\max}$ an upper bound for the number of nodes in $\mathcal{G}$. The number $\beta_{\max}$ is double exponential in the input size, which is for simplicity $d + r$. The value $\beta_{\max}$ is large enough to perform all computations over the semigroup $S$ and it is small enough in order to solve the reachability problem in the search graph $\mathcal{G}$ in non-deterministic space NSPACE($2^{2^{\mathcal{O}(d + r \log r)}}$). Using standard knowledge in complexity theory (like Savitch's Theorem, c.f. Hopcroft and Ullman (1979)), we can state:

**Theorem 5.7** *The satisfiability problem for word equations with regular constraints is in the following complexity classes:*

$$\mathit{DSPACE}\left(2^{2^{\mathcal{O}(d + r \log r)}}\right), \text{ i.e., double exponential deterministic space,}$$
$$\mathit{DTIME}\left(2^{2^{2^{\mathcal{O}(d + r \log r)}}}\right), \text{ i.e., triple exponential deterministic time.}$$

*The length of a shortest solution is at most four times exponential in the input size, it can be bounded by*

$$2^{2^{2^{2^{\mathcal{O}(d + r \log r)}}}}.$$

*The computation of a shortest solution by exhaustive search is possible in at most five times deterministic exponential time.*

**Remark 5.8** *The complexity bounds given above are slightly different from other bounds published in the literature so far. In Kościelski and Pacholski (1996: Cor. 4.6) a triple exponential non-deterministic time bound for the satisfiability problem is given. Here we have triple exponential deterministic time, since in the formulation as a graph reachability problem it is the* *number of nodes which becomes important. The upper bound for the exponent of periodicity given in Schulz (1992a) for the situation including regular constraints is based on the techniques of the original article of Makanin. This yields a double exponential bound whereas it is shown here that one exponential is enough. The upper bound for the exponent of periodicity is essentially optimal; the optimality is not known for the bounds mentioned in Thm. 5.7*

# 6 Notes

A systematic study of equations in free monoids was initiated by A. A. Markov in the late 1950's in connection with Hilbert's Tenth Problem, see Hmelevskiĭ (1971), Makanin (1981). It is not difficult to see that the matrices having non-negative integer coefficients and determinant 1 form a free monoid inside the special linear group $\mathrm{SL}_2(\mathbb{Z})$. The free generators are:

$$a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Let $L = R$ be a word equation over $\{a, b\}$ in unknowns $\Omega = \{x_1, \ldots, x_n\}$. Replace each variable $x_i \in \Omega$ by a matrix

$$\begin{pmatrix} \alpha_{i1} & \alpha_{i2} \\ \alpha_{i3} & \alpha_{i4} \end{pmatrix},$$

where $\alpha_{ij}$ denote variables over $\mathbb{N}$. Multiplying matrices corresponding to the words $L$ and $R$ yields an equation of the form

$$\begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix}.$$

The coefficients $P_1, \ldots, Q_4$ are polynomials in the $\alpha_{ij}$. It is clear that the equation $L = R$ has a solution if and only if the following Diophantine system has a non-negative solution:

$$\begin{aligned} P_i &= Q_i, \quad i = 1, \ldots, 4, \\ \alpha_{i1}\alpha_{i4} \Leftrightarrow \alpha_{i2}\alpha_{i3} &= 1, \quad\;\; i = 1, \ldots, n. \end{aligned}$$

The hope of Markov was to prove this way the unsolvability of Hilbert's Tenth Problem, which was not settled at that time. This failed: The unsolvability of Hilbert's Tenth Problem was shown in 1970 by Matiyasevich using quite different methods, see Matiyasevich (1993). The solvability of word equations is, needless to say at this place, due to Makanin (1977).

Before Makanin obtained the breakthrough only partial results were known. In 1964 and 1967 Hmelevskiĭ found a positive solution for the cases with two and three variables respectively, see Hmelevskiĭ (1971). In the case of two variables a polynomial time algorithm for the satisfiability problem is given in Charatonik and Pacholski (1993). The solvability in the case where each variable occurs at most twice is due to Matiyasevich (1968). Other special cases were solved in Plotkin (1972) and Lentin (1972). After the general solution was established in 1977 other questions became central. In Makanin (1979) it is shown that the rank of an equation is computable, see also Pécuchet (1981). Makanin's algorithm was implemented in 1987 at Rouen, see Abdulrab and Pécuchet (1990). The inherent complexity of the satisfiability problem of word equations is not known. The lower bound is NP-hardness for equations without regular constraints and PSPACE-hardness for equations with regular constraints. The known upper bounds to date for Makanin's algorithm are given in Thm. 5.7. There is a double exponential gap between lower and upper bound for the space complexity.

The original article Makanin (1977) is very technical. In the sequel other presentations with various improvements were given, let us refer to Jaffar (1990), Schulz (1992a, 1993). The present chapter is along this line, it is rather close to Schulz (1992a). A brief survey on equations in words can be found in Perrin (1989); more material on equations in free monoids and, especially on equations without constants, is in the Handbook of Formal Languages, see Choffrut and Karhumäki (1997). There are also two volumes in the Springer lecture notes series dedicated to word equations and related topics: Schulz (1992b) and Abdulrab and Pécuchet (1993).

Equations in free groups are defined analogously to word equations. The situation however becomes extremely complicated. It was Makanin himself who mastered also this problem. In Makanin (1982) and with a correction in Makanin (1984) it is shown that the satisfiability of group equations with constants in decidable. In Razborov (1984) an algorithm is presented which generates all solutions to a given equation. The inherent complexity of Makanin's algorithm for groups is investigated in Kościelski and Pacholski (1998). The authors define the notion of abstract Makanin algorithm. They show that this abstract scheme is not primitive recursive.

Another direction to extend Makanin's result is to include partial commutation: Let $I \subseteq \Sigma \times \Sigma$ be a relation between letters, which says when letters may commute (i.e., when they are *independent*). The quotient monoid $M(\Sigma, I) = \Sigma^*/\{ab = ba \mid (a,b) \in I\}$ is called the *free partially commutative monoid*. It was introduced in Cartier and Foata (1969), where interesting combinatorial properties were discovered. In computer science free partially commutative monoids are usually called *trace monoids*, a notion which is due to Mazurkiewicz (1977). The interest is that partial commutation expresses some basic phenomena of concurrency, let us refer to Diekert and Rozenberg (1995) for an overview. Syntactically, a system of trace equations is the same as a system of words equations, but solutions are searched in the trace monoid, this means the commutation relations $ab = ba$ can be used for free for all $(a,b) \in I$. For example, if $(a,b) \in I$, then, contrary to the situation in free monoids, the trace equation $axb \equiv bya$ has a solution $\sigma(x) = \sigma(y) = 1$. The set of all solutions of this trace equation is given by $\sigma(x) = \sigma(y)$ and alphabetic constraints.

It is shown in Matiyasevich (1997) that the satisfiability of a system of trace equations is decidable. The proof is a reduction of trace equations to word equations with regular constraints. As a byproduct of the reduction we may put arbitrary recognizable constraints on the variables without loosing satisfiability. Another reduction using a new result on lexicographic normal forms of traces is presented in Diekert, Matiyasevich, and Muscholl (1997).

A challenging open question to date is a generalization of Makanin's result to free partially commutative groups. But this is only one of many open questions in this area. The theory of word equations is still exciting and many problems remain to be solved.

## Problems

*Section 1*

*1.1* Decide whether or not the solution *abbababbaabab* given to Ex. 1.1 is a non-singular solution of minimal length.

*1.2* Show that the satisfiability problem of systems of word equations without regular constraints is NP-hard.
Hint: Show that the problem is NP-complete, if there is exactly one constant, $A = \{a\}$. Use the fact that linear integer programming is NP-hard, even in unary notation.

*1.3* Modify the decision procedure, where each variable occurs at most twice, to include the case where we have regular constraints. Show that the underlying decision procedure is PSPACE complete, if the regular constraints are specified by a list of NFA (non-deterministic finite automata).
Hint: The hardness follows directly from well-known PSPACE complete problems on regular sets.

*Section 2*

*2.1* Give a greedy algorithm to compute the $p$-stable normal form of a word $w \in A^*$. Modify the algorithm by pattern matching techniques such that it runs in linear time $\mathcal{O}(|w| + |p|)$.

*2.2* Prove Props. 2.8, 2.9, and 2.10. Show that the results remain true when there are regular constraints.

*2.3* Show that the satisfiability problem of single word equations without regular constraints is NP-hard.
Hint: Compare this problem with Prob. 1.2.

*2.4* Let $L_x \subseteq A^*$ be a regular language. Describe the set of all solutions $\sigma$ for an equation with only one unknown $x$ under the constraint $\sigma(x) \in L_x$.

*Section 3*

*3.1* An instance of a linear integer programming problem is given by an $m \times n$ matrix $D \in \mathbb{Z}^{m \times n}$ and a vector $c \in \mathbb{Z}^m$. Let $x \in \mathbb{N}^n$ be a minimal vector such that $Dx = c$. Assume that the sum over the squares over the coefficients in each row of $D$ is in $\mathcal{O}(1)$ and $\|c\| \in \mathcal{O}(n^2)$. Show by elementary methods that there is a (small) constant $c$ such that

$$\|x\| \in \mathcal{O}(2^{cn}).$$

Hint: The proof is a slight modification of the standard proof which shows that linear integer programming is NP-complete. Use Hadamard's Inequality for an upper bound for the maximal absolute value over the determinants of square submatrices of $D$. Next, show that if $x \in \mathbb{N}^n$ is a minimal solution, then there is also a minimal solution $x' \in \mathbb{N}^n$ such that first, the absolute value of at least one component can be bounded and second, $\sum_{i=1}^n x_i \le \sum_{i=1}^n x'_i$. Freeze by an additional equation one variable of $x'$

to be a constant. Repeat the process until the homogeneous system $Dx = c$ has only the trivial solution. Then apply Cramer's Rule.

It should be noted that this method doesn't yield the best possible result. But it is good enough to establish that $e(d) \in 2^{\mathcal{O}(d)}$, which was used in the proof of Thm. 3.2.

*Section 4*

*4.1* Consider the reduction in the proof of Lem. 4.14. Give an estimation for the length $d$ of the word equation and thereby for an upper bound of $e(\mathcal{B})$. Define another reduction where the denotational length of the resulting word equation becomes smaller. This improves also the estimation for $e(\mathcal{B})$. Give a third estimation for $e(\mathcal{B})$ based on the techniques presented in Sect. 3.

Hint to the second part: If a system contains two equations $x = x'$ and $xy = x'y'$, then the second one can be replaced by $y = y'$.

*4.2* The lower bound for $e(c(S), d)$ given in Ex. 3.4 can be refined. Kościelski and Pacholski (1996: Thm. 4.8) consider the following equation with $k = 5$:

$$x_n a x_n b x_{n-1} b \cdots x_2 b x_1 = a x_n x_{n-1}^k b x_{n-2}^k b \cdots x_1^k b a.$$

Show that there is a unique solution. Derive from this solution a lower bound for the constant hidden in the notation $e(c(S), d) \in c(S) \cdot 2^{\Omega(d)}$. Why is $k = 5$ a good value?

Hint: Show first that $\sigma(x_i) \in a^*$ for all $1 \leq i \leq n$.

## References

Abdulrab, H. and Pécuchet, J. (1990). Solving word equations, *J. Symbolic Computation*, *8*(5), 499–521.

Abdulrab, H. and Pécuchet, J. (Eds.). (1993). *Proceedings of Word Equations and Related Topics (IWWERT '91)*, Vol. 677 of *Lect. Notes Comp. Sci.*, Berlin-Heidelberg-New York. Springer-Verlag.

Cartier, P. and Foata, D. (1969). *Problèmes combinatoires de commutation et réarrangements.* No. 85 in Lecture Notes in Mathematics. Springer, Berlin-Heidelberg-New York.

Charatonik, W. and Pacholski, L. (1993). Word Equations with Two Variables, In Abdulrab, H. and Pécuchet, J.-P. (Eds.), *Proceedings of Word Equations and Related Topics, Second International Workshop, IWWERT'91, Rouen, France*, Vol. 677 of *Lect. Notes Comp. Sci.*, pp. 43–56 Berlin-Heidelberg-New York. Springer-Verlag.

Choffrut, C. and Karhumäki, J. (1997). Combinatorics of Words, In Rozenberg, G. and Salomaa, A. (Eds.), *Handbook of Formal Languages*, Vol. 1, pp. 329–438. Springer, Berlin-Heidelberg-New York.

Dickson, L. E. (1913). Finiteness of the odd perfect and primitive abundant numbers with $n$ distinct prime factors, *American Journal of Math.*, *35*, 413–422.

Diekert, V., Matiyasevich, Yu., and Muscholl, A. (1997). Solving trace equations using lexicographical normal forms, In Degano, P., Gorrieri, R., and Marchetti-Spaccamela, A. (Eds.), *Proc. of the 24th ICALP, Bologna, 1997*, No. 1256 in Lect. Notes Comp. Sci., pp. 336–347 Berlin-Heidelberg-New York. Springer-Verlag.

Diekert, V. and Rozenberg, G. (Eds.). (1995). *The Book of Traces*. World Scientific, Singapore.

Eyono Obono, S., Goralcik, P., and Maksimenko, M. (1994). Efficient Solving of the Word Equations in One Variable, In Privara, I. et al. (Eds.), *19th Symposium on Mathematical Foundations of Computer Science (MFCS'94), Košice (Slovakia) 1994*, No. 841 in Lect. Notes Comp. Sci., pp. 336–341 Berlin-Heidelberg-New York. Springer.

Gathen, J. von zur and Sieveking, M. (1978). A bound on solutions of linear integer equalities and inequalities, *Proceedings of the American Mathematical Society*, *72*(1), 155–158.

Hmelevskiĭ, Ju. I. (1971). Equations in Free Semigroups, In Petrovskiĭ, I. G. (Ed.), *Trudy Mat. Inst. Steklov. 107*. (In Russian) English translation in: Proceedings of the Steklov Institute of Mathematics 107 (1976). American Mathematical Society.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA.

Jaffar, J. (1990). Minimal and Complete Word Unification, *J. Assoc. Comput. Mach.*, *37*(1), 47–85.

Kościelski, A. and Pacholski, L. (1996). Complexity of Makanin's Algorithm, *J. Assoc. Comput. Mach.*, *43*(4), 670–684. Preliminary version in Proc. of the 31st Annual IEEE Symposium on Foundations of Computer Science, Los Alamitos (1990).

Kościelski, A. and Pacholski, L. (1998). Makanin's algorithm is not primitive recursive, *Theoret. Comput. Sci.*, *191*(1-2), 145–156.

Lentin, A. (1972). *Équations dans les monoïdes libres*. Gauthiers-Villars.

Lothaire, M. (1983). *Combinatorics on Words*, Vol. 17 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, MA.

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

Makanin, G. S. (1979). Recognition of the rank of equations in a free semigroup, *Izv. Akad. Nauk SSR*, *Ser. Mat. 43*. (In Russian) English translation in: Math. USSR Izvestija 14 (1980) 499–545.

Makanin, G. S. (1981). Equations in a free semigroup, *American Mathematical Society translations (2)*, *117*, 1–6.

Makanin, G. S. (1982). Equations in a free group, *Izv. Akad. Nauk SSR*, *Ser. Mat. 46*, 1199–1273. (In Russian) English translation in: Math. USSR Izvestija 21 (1983) 483–546.

Makanin, G. S. (1984). Decidability of the universal and positive theories of a free group, *Izv. Akad. Nauk SSR*, *Ser. Mat. 48*, 735–749. (In Russian) English translation in: Math. USSR Izvestija 25 (1985) 75–88.

Markowsky, G. (1977). Bounds on the index and period of a binary relation on a finite set, *Semigroup Forum*, *13*, 253–259.

Matiyasevich, Yu. (1968). A connection between systems of word and length equations and Hilbert's Tenth Problem, *Sem. Mat. V. A. Steklov Math. Inst. Leningrad*, *8*, 132–144. (In Russian) English translation in: Seminars in Mathematics, V. A. Steklov Mathematical Institute 8 (1970) 61–67.

Matiyasevich, Yu. (1993). *Hilbert's Tenth Problem.* MIT Press, Cambridge, Massachusetts.

Matiyasevich, Yu. (1997). Some Decision Problems for Traces, In Adian, S. and Nerode, A. (Eds.), *Proceedings of the 4th International Symposium on Logical Foundations of Computer Science (LFCS'97), Yaroslavl, Russia, July 6–12, 1997*, No. 1234 in Lect. Notes Comp. Sci., pp. 248–257 Berlin-Heidelberg-New York. Springer-Verlag.

Mazurkiewicz, A. (1977). Concurrent Program Schemes and their Interpretations, DAIMI Rep. PB 78, Aarhus University, Aarhus.

Pécuchet, J.-P. (1981). Sur la détermination du rang d'une équation dans le monoïde libre, *Theoret. Comput. Sci.*, *16*, 337–340.

Perrin, D. (1989). Equations in words, In Ait-Kaci, H. and Nivat, M. (Eds.), *Resolution of equations in algebraic structures, Vol. 2*, pp. 275–298. Academic Press.

Plotkin, G. (1972). Building in equational theories, *Machine Intelligence*, *7*, 115–162.

Razborov, A. A. (1984). On systems of equations in a free group, *Izv. Akad. Nauk SSR*, *Ser. Mat. 48*, 779–832. (In Russian) English translation in: Math. USSR Izvestija 25 (1985) 115–162.

Schulz, K. U. (1992a). Makanin's Algorithm for Word Equations: Two Improvements and a Generalization, In Schulz, K.-U. (Ed.), *Proceedings of Word Equations and Related Topics, 1st International Workshop, IWWERT'90, Tübingen, Germany*, Vol. 572 of *Lect. Notes Comp. Sci.*, pp. 85–150 Berlin-Heidelberg-New York. Springer-Verlag.

Schulz, K. U. (Ed.). (1992b). *Proceedings of Word Equations and Related Topics (IWWERT '90)*, Vol. 572 of *Lect. Notes Comp. Sci.*, Berlin-Heidelberg-New York. Springer-Verlag.

Schulz, K. U. (1993). Word Unification and Transformation of Generalized Equations, *Journal of Automated Reasoning*, *11*(2), 149–184.

# Index