CHAPTER 12

Makanin's Algorithm

12.0. Introduction

A seminal result of Makanin 1977 states that the existential theory of equations over free monoids is decidable. Makanin achieved this result by presenting an algorithm which solves the satisfiability problem for word equations with constants. The satisfiability problem is usually stated for a single equation, but this is no loss of generality.

This chapter provides a self-contained presentation of Makanin's result. The presentation has been inspired by Schulz 1992a. In particular, we show the result of Makanin in a more general setting, due to Schulz, by allowing that the problem instance is given by a word equation $L = R$ together with a list of rational languages $L_x \subseteq A^*$, where $x \in \Omega$ denotes an unknown and A is the alphabet of constants. We will see that it is decidable whether or not there exists a solution $\sigma : \Omega \to A^*$ which, in addition to $\sigma(L) = \sigma(R)$, satisfies the rational constraints $\sigma(x) \in L_x$ for all $x \in \Omega$. Using an algebraic viewpoint, rational constraints mean to work over some finite semigroup, but we do not need any deep result from the theory of finite semigroups. The presence of rational constraints does not make the proof of Makanin's result much harder; however, the more general form is attractive for various applications.

In the following we explain the outline of the chapter; for some background information and more comments on recent developments we refer to the Notes.

The major step toward Makanin's result is to bound the exponent of periodicity, which is, by definition, the maximal number of direct repetitions of a primitive word in a solution of minimal length. A priori it is not at all clear why an upper bound for the exponent of periodicity is a key, since there are arbitrarily long words where the exponent of periodicity is 3. This means that the exponent of periodicity alone

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

does not give any recursive bound on the length of a minimal solution. However, together with a deep combinatorial analysis in the situation of word equations, it does. The bound for the exponent of periodicity is calculated in Section 12.2 using the notion of p-stable normal form (Subsection 12.1.5) and some standard linear algebra.

Instead of working with word equations directly, it turns out to be more convenient to work with boundary equations. Systems of boundary equations are introduced in Section 12.3. In some sense they store the relative lengths of the variables in possible solutions. The important point is that a notion of convex chain can be defined (Subsection 12.3.4). This leads to a geometrical reflection on the problem. An upper bound for the exponent of periodicity yields an upper bound on the maximal length of clean convex chains (Proposition 12.3.15). As long as the convex chain condition is satisfied, the maximal length of convex chains yields an upper bound on the number of boundary equations (Corollary 12.3.16). The strategy of Makanin's algorithm is therefore as follows. A word equation is transformed into a system of boundary equations, which will satisfy the convex chain condition for trivial reasons. Then transformation rules are applied which maintain the convex chain condition (which is not trivial) and which either lead to a solution of the word equation or introduce more and more boundary equations. But, for the number of boundary equations, there is an upper bound provided by the exponent of periodicity. Hence, we can stop the procedure at some stage. The transformation rules (Subsection 12.3.5) are at the heart of Makanin's algorithm. The central idea is a left-to-right transport of positions in combination with a splitting of variables. It is not so much Makanin's algorithm which is complicated; the hard part is the termination proof, when to stop the procedure. Major steps are Proposition 12.3.15 and the proof that the transformations preserve the convex chain condition. Makanin's algorithm itself becomes the construction of a finite search graph: the vertices are systems of boundary equations and edges are transformation rules.

During our presentation we do not focus on necessary decidable conditions which might be used to prune the search graph. A good pruning strategy is of course extremely important for an implementation since the search graph tends to be huge. However, pruning doesn't help to understand the algorithm, nor does it seem to have any effect on the worst-case analysis. For the worst-case analysis we use standard notions of complexity theory as they can be found in the textbooks of Hopcroft and Ullman 1979 and Papadimitriou 1994. The final result of this chapter shows that Makanin's algorithm can be implemented in exponential space, Theorem 12.4.2.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Exponential space is not optimal for the satisfiability problem of word equations, since Plandowski 1999b has shown that the satisfiability problem of word equations can be decided in polynomial space. Plandowski's new approach is rather different from the material presented here; for example, an important ingredient of Plandowski's method is data compression in terms of exponential expressions, whereas we do not need any data compression here. Makanin's algorithm has many other nice features, and, since the equations are written in plain form, it seems to be easier to follow some strategy during the search for a solution. Experimental results indicate Makanin's algorithm is quite suitable for practical application.

12.1. Words and word equations

12.1.1. Basic notions

By $A = \{a, b, \ldots\}$ we mean an alphabet of constants and Ω is a set of *variables* (or *unknowns*) such that $A \cap \Omega = \emptyset$. Throughout this chapter we shall use the same symbol σ to denote a mapping $\sigma : \Omega \to A^*$ and its canonical extension to a homomorphism $\sigma : (A \cup \Omega)^* \to A^*$ leaving the letters of A invariant. The empty word (and also the unit element in other monoids) is denoted by ε. The length of a word w is denoted by $|w|$. We have $|\varepsilon| = 0$. The prefix relation of words is denoted by $u \leq v$, the proper prefix relation is $u < v$. As usual, the set of integers is $\mathbb{Z}$. The set of natural numbers is $\mathbb{N}$; these are the nonnegative integers. Lower-case Greek letters like α, β etc. are mostly used to denote natural numbers. By $\log \alpha$ we mean $\max\{1, \lceil \log_2 \alpha \rceil\}$.

A *word equation* is a pair $(L, R) \in (A \cup \Omega)^* \times (A \cup \Omega)^*$; it is written as $L = R$. A *system* of word equations is a set of equations $\{L_1 = R_1, \ldots, L_k = R_k\}$. A system where each variable occurs at most twice is called a *quadratic system.* 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 $1 \leq i \leq k$. It is called *nonsingular*, if $\sigma(x) \neq \varepsilon$ for all $x \in \Omega$; otherwise it is called *singular.*

Example 12.1.1. Let $A = \{a, b\}$ and $\Omega = \{x, y, z, u\}$. Consider the equation

$$xauzau = yzbxaaby.$$

This is a solvable quadratic equation. There are singular and nonsingular solutions. A possible nonsingular solution is given by

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

We have

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

12.1.2. Solving quadratic systems

Using Nielsen transformations there is a simple strategy for solving quadratic systems. The strategy is as follows. Let $E = \{L_1 = R_1, \ldots, L_k = R_k\}$ be a system of word equations and assume that every variable $x \in \Omega$ occurs at most twice in the system. Let $\|E\| = \sum_{i=1}^{k} |L_i R_i|$ denote the denotational length of E. Using induction on $\operatorname{Card}\Omega$ we describe a nondeterministic decision algorithm which solves the question whether there is a solution in space $O(\|E\|)$. The case $\Omega = \emptyset$ is trivial, hence let $\Omega \neq \emptyset$. The first step is the guess whether there is a solution $\sigma : \Omega \to A^*$ such that $\sigma(x) = \varepsilon$ for some $x \in \Omega$. This is done by choosing some $\Omega' \subseteq \Omega$ and by replacing all occurrences of all $x \in \Omega'$ by the empty word. We obtain a new system E' over $\Omega \setminus \Omega'$ and recursively, if $\Omega' \neq \emptyset$, we decide in nondeterministic linear space whether E' has a solution. Thus, after this step we are looking for nonsingular solutions of E, only. We may assume that the first equation is of the form

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

By symmetry (or a nondeterministic guess to interchange the roles of L_1 and R_1) we may write either $x = az$ or $x = yz$, where z is a new variable. Replacing the 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 also occurs 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 $\|E'\| \leq \|E\|$. Clearly, if E has a nonsingular solution, then E' is solvable including the possibility of a singular solution with $\sigma(z) = \varepsilon$. However, if E' is solvable, then E is also solvable. Now, let $\sigma : \Omega \to A^*$ be a nonsingular solution of E where $\sum_{x \in \Omega} |\sigma(x)|$ is minimal. Then we find a solution σ' for E' with $|\sigma'(z)| < |\sigma(x)|$ since $\sigma(y) \neq \varepsilon$. Thus, the length of a shortest solution has decreased. This shows that the nondeterministic procedure will find a solution, if there is any. The space requirement for this algorithm is linear, but its time complexity might be exponential. The exponential time bound is perhaps inevitable, because the satisfiability problem for quadratic word equations remains NP-hard.

The algorithm above has a convenient graphical representation which we show by another example: Consider $A = \{a, b, c\}$ and $\Omega = \{x, y, z\}$.

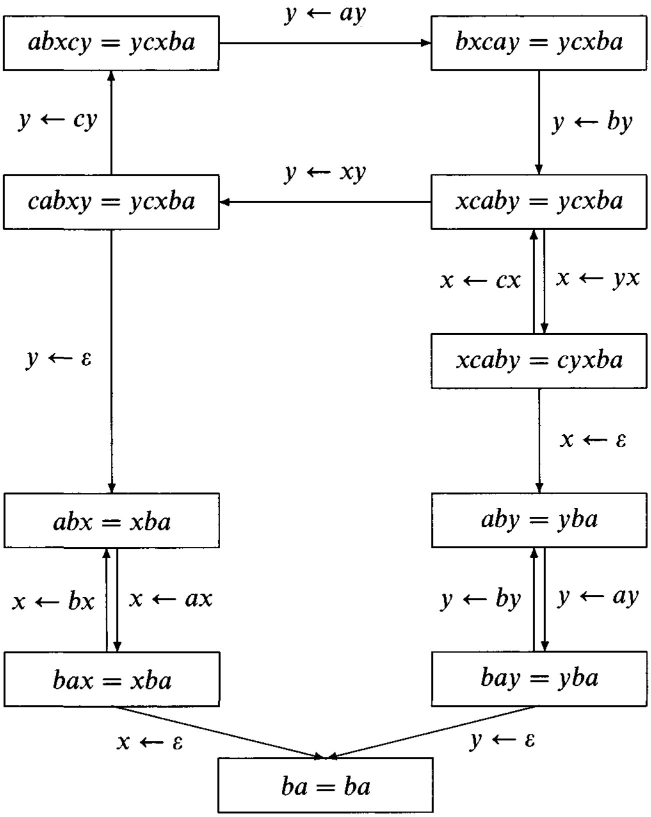


Figure 12.1. Solving the equation $abxcy = ycxba$.

Let the word equation be $abxcy = ycxba$. Running the algorithm leads to the graph as depicted in Figure 12.1. The arcs are labeled in such a way that we can reconstruct a solution by going backwards on a path from the initial equation to the trivial equation $ba = ba$. One of the paths has the following labels:

$$y \leftarrow ay,\ y \leftarrow by,\ x \leftarrow yx,\ x \leftarrow \varepsilon,\ y \leftarrow ay,\ y \leftarrow \varepsilon.$$

It corresponds to the minimal solution, where $\sigma(x) = a$ and $\sigma(y) = aba$. Nodes or arcs which cannot lead to any solution have been omitted in the picture; they are not drawn.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

12.1.3. Combinatorial properties

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

Proposition 12.1.2. *Let $x, y, z \in A^*$ be words, $y, z \neq \varepsilon$. Then the following assertions are equivalent:*

(i) $xy = zx$,

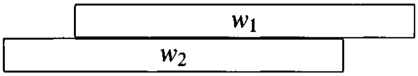
(ii) $\exists\, r, s \in A^*, s \neq \varepsilon, \alpha \geq 0 : x = (rs)^\alpha r,\ y = sr$, *and* $z = rs$.

A word p is called *primitive*, if it cannot be written in the form $p = r^\alpha$ with $r \in A^+$ and $\alpha \neq 1$. In particular, a primitive word p is nonempty, $p \neq \varepsilon$.

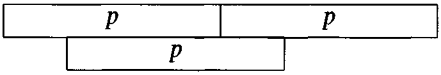
Proposition 12.1.3. *Let $p \in A^*$ be primitive and $p^2 = xpy$ for some $x, y \in A^*$. Then we have either $x = \varepsilon$ or $y = \varepsilon$ (but not both).*

Proofs of Propositions 12.1.2 and 12.1.3 can be found e.g. in Lothaire 1983, Section 1.3.

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 $x \neq \varepsilon$ and sometimes the figure also indicates that both $y \neq \varepsilon$ and $z \neq \varepsilon$. But there will be no risk of confusion. For example, Proposition 12.1.3 can be rephrased by saying that the following picture is not possible for a primitive word.

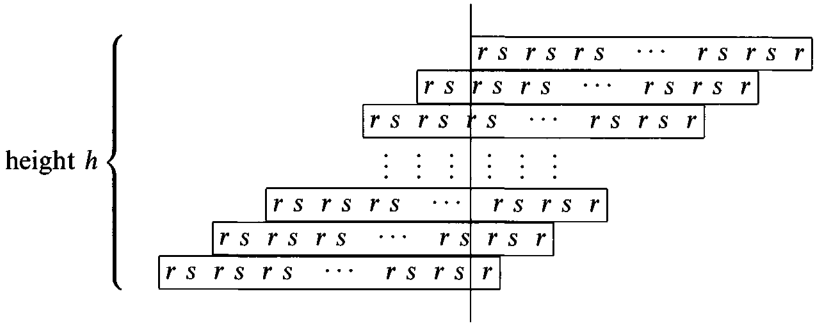


12.1.4. Domino towers

Every nonempty word $w \in A^+$ can be written in the form $w = (rs)^{h-1}r$ with $s \neq \varepsilon$, $h \geq 2$. This is trivial for $r = \varepsilon$ and $h = 2$. The more interesting case is when we have $r \neq \varepsilon$. Writing $w = (rs)^{h-1}r$ leads to an arrangement

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

of the following shape:

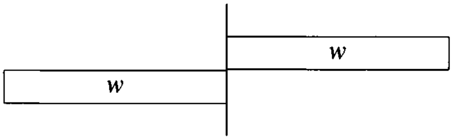

height h
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$
$r\ s\ r\ s\ r\ s\ \cdots\ r\ s\ r\ s\ r$

The position of the vertical line says that the upper left boundary is never to the right of the lower right boundary. The formal definition of such an arrangement also allows a less uniform shape. Let $h \geq 2$. We say that a nonempty 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 nonempty words $y_1, \ldots, y_{h-1}$, $z_2, \ldots, z_h \in A^+$, such that

(i) $w = x_i y_i = z_{i+1} x_i$ for all $1 \leq i < h$,

(ii) $|z_2 \cdots z_h| \leq |w|$.

In the figure above we have $x_1 = \cdots = x_{h-1} = (rs)^{h-2} r$, $y_1 = \cdots = y_{h-1} = sr$, and $z_2 = \cdots = z_h = rs$. Note also that a domino tower of height 2 may degenerate as in the following figure.



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

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

LEMMA 12.1.4. *Let $h \geq 2$ and $w \in A^+$ be a nonempty word which can be arranged in a domino tower of height h. Then we have* $\exp(w) \geq h-1$.

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 $(h-1)|z| \leq |w|$, and we have $xy = zx = w$. Hence y and z are conjugate and we may apply Proposition 12.1.2. We obtain $z = rs$ and $x = (rs)^\alpha r$ for some $\alpha \geq 0$ and $|r| < |z|$. Hence $w = z^{\alpha+1} r$ and therefore

$$(h-1)|z| \leq |w| < (\alpha+2)|z|.$$

Since $|z| > 0$ we see that $h - 1 \leq \alpha + 1 \leq \exp(w)$. ■

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

12.1.5. Stable normal forms

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

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

such that $k \geq 0$, $u_0, u_i \in A^*$, $\alpha_i \geq 0$ for $1 \leq i \leq k$, and the following three conditions are satisfied:

(i) $w = u_0 p^{\alpha_1} u_1 \cdots p^{\alpha_k} u_k$;

(ii) $k = 0$ if and only if p^2 is not a factor of w;

(iii) if $k \geq 1$, then

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

Example 12.1.5. Let $p = aba$ and $w = ab(aba)^5ba(aba)^4ba$. Then the p-stable normal form of w is the sequence

$$(ababa, 3, ababa, 3, ababa).$$

Proposition 12.1.6. *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 $1 \leq i \leq k$.*

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 ℓ are both minimal, hence $k = \ell$.

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

We show first that $u_0 = v_0$. To see this, suppose by symmetry that $|u_0| \leq |v_0|$. Since $u_0 p \in A^*p^2$ and $v_0 \in (A^*p \setminus A^*p^2A^*)$, we obtain that $u_0 \leq v_0 < u_0 p$. By Proposition 12.1.3 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 $u_1 \neq p$, Proposition 12.1.3, 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 that $(u_1, \alpha_2, u_2, \ldots, \alpha_k, u_k) = (v_1, \beta_2, v_2, \ldots, \beta_k, v_k)$. ∎

12.1.6. The existential theory of concatenation

The existential theory of equations over free monoids is decidable, i.e., the satisfiability of any propositional formula over word equations (with rational constraints) can be decided. This can be deduced from Makanin's result as follows. 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 12.1.7. *An inequality $L \neq R$ is equivalent with the following positive existential formula:*

$$\exists x \exists y \exists z : \bigvee_{a \in A} (L = Rax \vee R = Lax) \vee \bigvee_{a,b \in A,\, a \neq b} (L = xay \wedge R = xbz).$$

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 Proposition 12.1.8. It relies on the observation 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 12.1.8. *Let $a, b \in A$ be distinct letters, $a \neq b$, 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 with the set of solutions of the following equation:*

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

Sometimes it is useful to do the opposite of Proposition 12.1.8 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 12.1.9. *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 is in canonical bijection with the set of solutions of the following system:*

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

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

It is worth noting that a disjunction of word equations can be replaced by an existential formula in a single equation, too. The construction shown below has been taken from Karhumäki, Mignosi, and Plandowski 2000.

Proposition 12.1.10. *Let $a, b \in A$ be distinct letters, $a \neq b$. A disjunction of two word equations is equivalent with a single word equation in two extra unknowns.*

Proof. Consider a disjunction

$$L_1 = R_1 \vee L_2 = R_2,$$

where $L_1, L_2, R_1, R_2 \in (A \cup \Omega)^*$. This is equivalent to the disjunction

$$L_1 R_2 = R_1 R_2 \vee R_1 L_2 = R_1 R_2.$$

Thus, for the construction we can start with a disjunction where the right-hand sides are equal: $L_1 = R \vee L_2 = R$.

It turns out that the word $P = L_1 L_2 R a L_1 L_2 R b \in (A \cup \Omega)^+$ is primitive. In fact, we need a sharper statement: Choose a primitive word $Q \in (A \cup \Omega)^+$ and some $\alpha \geq 1$ such that P is a prefix of Q^α. Then we have $|Q| > \frac{1}{2}|P|$. To see this, assume to the contrary that $|Q| \leq \frac{1}{2}|P|$. Since $a \neq b$ we have $|Q| < \frac{1}{2}|P|$ and Q is a prefix of $L_1 L_2 R$. But this is impossible due to Proposition 12.1.3. As a consequence, if P^2 is a factor of some word $P^2 W P^2$ where $|W| \leq \frac{1}{2}|P|$, then P^2 is either a prefix or a suffix of $P^2 W P^2$. (This can be seen from the statement above, again using Proposition 12.1.3, and by Proposition 12.1.2.)

Given this, let x, y be two extra unknowns. Then the disjunction $L_1 = R \vee L_2 = R$ is equivalent to the existential formula

$$\exists x \exists y : P^2 L_1 P^2 L_2 P^2 = x P^2 R P^2 y.$$

Indeed, if $L_1 = R \vee L_2 = R$ is solvable then we can satisfy the existential formula above. For the other direction let σ be a solution to the formula. Since $|\sigma(L_i)| \leq \frac{1}{2}|\sigma(P)|$, $i = 1, 2$, the first P^2 of the right-hand side matches either the first or the second P^2 on the left-hand side. If it matches the second one, we are done. Hence we may assume that the first P^2 of the right-hand side matches the first P^2 on the left-hand side. Now the second P^2 of the right-hand side cannot match the third P^2 on the left-hand side since $|\sigma(R)| < |\sigma(L_1 P^2 L_2)|$, so it matches the second one. The assertion follows. ■

Let us look at the number of different constants which are used in a word equation. It is well known that the problem of solving word equations can be reduced to the case where only two constants appear:

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Proposition 12.1.11. *Let $L = R$ be a word equation over a set of constants A and $B = \{a,b\}$ be a two-letter alphabet. Then we can construct (in polynomial time) a word equation over B which is solvable if and only if $L = R$ has a nonsingular solution.*

Proof. We may assume that $A = \{a_1,\dots,a_k\}$ with $k > 2$. We define an injective homomorphism $\eta : (A \cup \Omega)^* \longrightarrow (B \cup \Omega)^*$ by $\eta(a_i) = ab^ia$ for $1 \le i \le k$ and $\eta(x) = axa$ for $x \in \Omega$. We obtain an equation $\eta(L) = \eta(R)$.

Clearly, if $L = R$ has a nonsingular solution $\sigma : \Omega \longrightarrow A^+$, then for all $x \in \Omega$ we can write $\eta(\sigma(x)) = a\tau(x)a$; and $\tau : \Omega \longrightarrow B^+$ is a nonsingular solution of $\eta(L) = \eta(R)$.

For the converse, let $\tau : \Omega \longrightarrow B^*$ be any solution of $\eta(L) = \eta(R)$. (Even if this solution is singular, we will produce a nonsingular solution of $L = R$.) Define $\sigma'(x) = a\tau(x)a$ for $x \in \Omega$, and modify η by defining $\eta'(a_i) = ab^ia$ and $\eta'(x) = x$ for $1 \le i \le k$ and $x \in \Omega$. Let $L' = \eta'(L)$, and $R' = \eta'(R)$. Then σ' is a nonsingular solution of $L' = R'$ such that $\sigma'(x) \in aB^*a$ for all $x \in \Omega$. Of course, we cannot guarantee that $\sigma'(x) \in \eta(A)^+$, it might happen that $\sigma'(x)$ contains factors of the form aaa or ab^+ab^+a or so. But such a *wrong* factor on one side of the equation must correspond to the same wrong factor on the other side, which must be again inside some piece corresponding to a variable. In order to formalize this idea we observe that the subset $\{\varepsilon\} \cup (aB^* \cap B^*a)$ is a free submonoid of B^*. The (infinite) basis is $\Sigma = \{a\} \cup aB^*a \setminus B^*aaB^*$. Hence $\sigma' \circ \eta' : \Omega \longrightarrow \Sigma^+$ is a nonsingular solution of the original equation $L = R$ if we identify $\eta'(A)$ with A. The only difference is that $\sigma'(x)$ may contain (finitely many) letters from $\Sigma \setminus \eta'(A)$. Hence for some finite set $C \subseteq \Sigma \setminus \eta'(A)$ we have $\sigma'(LR) \subseteq (\eta'(A) \cup C)$. Choosing any mapping $\rho : C \longrightarrow \eta'(A)^+$ we obtain a nonsingular solution $\sigma = \rho \circ \sigma'$, which can be identified with a nonsingular solution of $L = R$ using the following composition leaving the letters of A invariant:

$$(A \cup \Omega) \xrightarrow{\eta'} (\eta'(A) \cup \Omega)^* \xrightarrow{\sigma'} (\eta'(A) \cup C)^+ \xrightarrow{\rho} \eta'(A)^+ \xrightarrow{\eta'^{-1}} A^+. \qquad \blacksquare$$

12.1.7. 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.

Let E be a set of word equations where exactly one variable x occurs, $\Omega = \{x\}$. By Proposition 12.1.8 we may assume that E is given by a single equation $L = R$ with $L, R \in (A \cup \{x\})^*$. The basic check is whether $\sigma(x) = \varepsilon$ yields the singular solution. It is therefore enough to consider

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

only nonsingular solutions. Let us denote by $\mathscr{L}$ a list of pairs (p, r) where $p \in A^+$ is primitive and $r \in A^*$ is some prefix $r < p$. We say that $\mathscr{L}$ is *complete for the equation* $L = R$, if every nonsingular solution σ has the form $\sigma(x) = p^\alpha r$ for some $\alpha \geq 0$ and $(p, r) \in \mathscr{L}$.

Assume for a moment that a finite complete list $\mathscr{L}$ has already been computed in a first phase of the algorithm. Then we proceed as follows. For each pair $(p, r) \in \mathscr{L}$ 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 α now denotes an integer variable. Thus, the problem is now to find solutions for α such that $\alpha \geq 2$. Using the symbolic expression we can factorize L and R in their p-stable normal forms:

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

Here $k, \ell \geq 0$ 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 Proposition 12.1.6 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:

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

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

It is clear that for each pair (p, r) the necessary computations can be done in polynomial time. In fact, using pattern-matching techniques it can be proved that linear time is enough for each pair (p, r). The performance of the algorithm therefore depends on an efficient computation of a short and complete list $\mathscr{L}$.

We may assume that $L = ux \cdots$ and $R = xv \cdots$, 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 σ is a solution of $L = R$, then σ also solves an equation of type $ux = xw$ for some word $w \in A^+$. By Proposition 12.1.2 it is immediate that we have $\sigma(x) = p^\alpha r$ for some $\alpha \geq 0$ and $r < p$. Thus, the obvious method is to define the list $\mathscr{L}$ by all pairs (p, r) where $r < p$. We obtain a list $\mathscr{L}$ with $|p|$ elements.

There is an improvement of the algorithm due to Eyono Obono, Goralčik, and Maksimenko 1994 by observing that there is a complete list $\mathscr{L}$ of at most logarithmic length. This improvement uses a finer combinatorial analysis and it relies, in particular, on the following well-known fact:

- Let $u, v, w \in A^+$ be primitive words such that $u^2 < v^2 < w^2$. Then we have $|u| + |v| \leq |w|$. In particular, a word $w \in A^*$ of length n has at most $O(\log n)$ distinct prefixes of the form pp where p is primitive.

For a proof of the fact see Lemma 8.1.14 or Crochemore and Rytter 1995, Lemma 10.

We outline the method of Eyono Obono et al. 1994: The set of nonsingular solutions is divided into two classes. The first class contains all solutions where $|\sigma(x)| \geq |u| - |v|$. (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 σ is a solution with $|\sigma(x)| \geq |u| - |v|$, 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 conjugate, then there is no such solution. Otherwise, if p and q are conjugate, we include the unique pair (p, r) into $\mathscr{L}$. This pair covers all solutions where $|\sigma(x)| \geq |u| - |v|$.

Now, let σ be a nonsingular solution such that $0 \neq |\sigma(x)| < |u| - |v|$. 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 nonempty 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 also is 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 $v = \varepsilon$, then we add all pairs (q, ε) to $\mathscr{L}$. Otherwise, if $v \neq \varepsilon$, then we compute for each q the unique factorization $q = sr$ with $s \neq \varepsilon$ such that $v \in q^*s$. We add all pairs (rs, r) to $\mathscr{L}$. It follows from Crochemore 1981 that the list $\mathscr{L}$ can be computed in time $O(|LR| \log |LR|)$. The conclusion is that the solvability of an equation $L = R$ in one variable can be decided in time $O(|LR| \log |LR|)$. It is however not clear whether there is a linear-time algorithm.

12.1.8. Constraints over a semigroup

The input for Makanin's algorithm is an equation $L = R$ with $L, R \in (A \cup \Omega)^*$ together with rational languages $L_x \subseteq A^*$ for all variables $x \in \Omega$. We assume that the languages are specified by nondeterministic finite automata. If it happens that for some variable no rational constraint is defined, then we simply put $L_x = A^*$. 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, henceforth we will not distinguish between variables and constants in the equation. Every constant $a \in A$ is replaced by a new

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

variable x_a and the constraint $L_{x_a} = \{a\}$ for all $a \in A$. (For readability we shall 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 nonsingular solutions. Hence we shall assume that $\varepsilon \notin L_x$ 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 purposes we require that φ 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^ε the monoid which is obtained by adjoining a unit element ε to S. We have $S^\varepsilon \setminus \{\varepsilon\} = S$ and the homomorphism φ is extended to a monoid homomorphism $\varphi : A^* \to S^\varepsilon$. We have $\varphi^{-1}(\varepsilon) = \{\varepsilon\}$ 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 $s \in S^\varepsilon$. 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 $s \in S$ and $r \geq 0$ and $\alpha, \beta \geq 1$.

REMARK 12.1.12. Assume that each rational language L_x is specified by a (nondeterministic) finite automaton with r_x states, $x \in \Omega$. Let $r = \sum_{x \in \Omega} r_x$. Then we may choose the semigroup S such that

$$\operatorname{Card} S \leq 2^{r^2} \text{ and } c(S) \leq r!$$

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 Card S and $c(S)$ are not relevant. They are used only later (Subsection 12.4.2) when complexity issues are investigated.

12.2. The exponent of periodicity

This section provides an effective upper bound for the exponent of periodicity in a solution of minimal length of a given word equation (with rational constraints). For the decidability result any effective upper bound would be sufficient, but, due to its close relation to linear Diophantine equations and by techniques from linear optimization, one can be precise.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

The upper bound for the exponent of periodicity is exponential in the input size, and this is essentially optimal. In the proof below, a rather detailed analysis is given hiding perhaps some basic ideas. In a first reading one is therefore invited to ignore the exact values. We shall use the notations as introduced in Subsection 12.1.8.

THEOREM 12.2.1. *Let $d \geq 1$ be a natural number, $\varphi: A^* \to S^\varepsilon$ a homomorphism, and $c(S) \geq 2$ as above. There is a computable number $e(c(S), d) \in c(S) \cdot 2^{O(d)}$ 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:

(i) $\varphi\sigma'(x) = \varphi\sigma(x)$ *for all* $x \in \Omega$,

(ii) $w = \sigma(x_1 \cdots x_g) = \sigma(x_{g+1} \cdots x_d)$,

(iii) $\exp(w) \leq e(c(S), d)$.

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 σ 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 σ 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:

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

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

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

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Since w is a solution, there are many equations among the words and among the integers. For example, for $k, \ell \geq 2$ 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

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

We have no bound on k, ℓ, or m, but we have $|k + \ell - m| \leq 2$. 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 that $w_{k+1} = abp$ 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 $r_k < c(S) - 1$, then $\alpha_k = \gamma_k$, otherwise $\alpha_k + 1 = \gamma_k$. Similarly, if $s_1 < c(S) - 1$, 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 $c(S) \geq 2$, the case $u_k v_0 = p^3$ leads to the equation

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

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 $\gamma - (\alpha + \beta) = c$ 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 nontrivial

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

equation having at most three variables. Proceeding this way through all $d-2$ word equations we have various interactions due to renaming and substitution. However, finally each equation $xy = z$ leads to at most one nontrivial equation with at most three variables. The type of this equation is

$$c_1\gamma + i_1 - c_2\alpha - i_2 - c_3\beta - i_3 = c$$

where we have $0 \le i_1, i_2, i_3 \le d-2$, $0 \le c \le 2$, $c_1, c_2, c_3 \in \{0,1\}$. This can be written as

$$c_1\gamma - c_2\alpha - c_3\beta = c' \text{ with } |c'| \le 2d-2.$$

For the case $\alpha = \beta \ne \gamma$ and $c_1 = c_2 = c_3 = 1$ we obtain a coefficient -2, because then $\gamma - 2\alpha = c'$.

We have viewed the symbols $\alpha, \beta, \ldots$ as variables ranging over natural numbers. Going back to the solution σ, which is given by the word w, the symbols $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_\ell, \gamma_1, \ldots, \gamma_m$ represent concrete values. 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 under φ. If $\delta \ge 1$ is a remaining value, i.e., a number greater than zero, then we replace it by $\delta = 1 + Z_\delta$ where now Z_δ denotes a variable over $\mathbb{N}$. For example an equation

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

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

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

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

What becomes crucial now is the converse: Every solution in nonnegative 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 $\mathbb{N}^n$.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

In this ordering we have $(\alpha_1,\ldots,\alpha_n) \le (\beta_1,\ldots,\beta_n)$ if and only if $\alpha_i \le \beta_i$ for all $1 \le i \le n$.

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

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

Obviously, there are only finitely many systems of linear Diophantine equations where the numbers of equations, variables, and the absolute values of coefficients and constants are bounded. For each system the set of minimal solutions is finite; this is a special case of Lemma A of 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 $e(d)+d-1 \ge \alpha_1,\ldots,\beta_1,\ldots$ 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 is known that $e(d) \in 2^{O(d)}$; see Remark 12.2.2. Hence we can state

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

This proves the theorem. ■

Remark 12.2.2. The result on the exponent of periodicity $e(d)$ saying that it can be bounded by a singly exponential function is due to Kościelski and Pacholski 1996. The analysis given there is more accurate than the one presented here, and it leads to linear Diophantine systems having a slightly different structure. The article uses results of 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 (without rational constraints) is in $O(2^{1.07d})$. The introduction of rational constraints doesn't change the situation very much: it yields the factor $c(S)$, as is shown above. Therefore the actual result including rational constraints is

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

It is rather difficult to obtain this very good bound. However, a bound which is good enough to establish Theorem 12.2.1 is $e(d) \in O(2^{cd})$

for some constant c, say $c = 4$. Such a more moderate bound can be obtained using the present approach and some standard knowledge in linear algebra; see Problem 12.3.1.

Example 12.2.3. Consider $c, n \geq 2$ and let $S = \mathbb{Z}/c\mathbb{Z}$ be the cyclic group of c elements. We give a rational 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 σ is $\sigma(x_i) = a^{c \cdot 2^{i-1}}$, $1 \leq i \leq n$. A transformation into a single equation according to Proposition 12.1.8 shows that $e(c(S), d) \in c(S) \cdot 2^{\Omega(d)}$. Thus, the assertion given in Theorem 12.2.1 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 12.2.4. Consider the following system of word equations:

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

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 12.2.5. Consider the equation with rational constraints

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

A suitable homomorphism $\varphi\colon \{a,b\}^+ \to S$ 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:

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

The constant $c(S) = 2$ fits the requirement $s^{r+c(S)} = s^{r+\alpha c(S)}$ for all $s \in S^{\varepsilon}$ and $r \geq 0$, $\alpha \geq 1$. It is not difficult to find a solution σ for the equation above, e.g. $\sigma(x) = a^2$, $\sigma(y) = ba^2$, and $\sigma(z) = a^3ba^2$. Now let α, β, γ, and δ 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 rational constraints.

12.3. Boundary equations

12.3.1. Linear orders over a semigroup

We introduce some concepts using the semigroup S which describes the rational constraints. 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 rational

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

constraints and that there is a nonsingular solution σ such that $\sigma(x_i) = u_i$ for $1 \le i \le d$. The equation and the solution define a word $w \in A^+$ and two factorizations $w = u_1 \cdots u_g = u_{g+1} \cdots u_d$. The positions between the factors u_i and u_{i+1} for $1 \le i < g$ or $g < i < d$ are called *cuts*. By convention, the first and the last position of w are also cuts, and then we have at most d cuts. Reading the word from cut to cut, we obtain a sequence $(w_1, \ldots, w_m)$ such that each u_i is a product of some w_k and such that $w = w_1 \cdots w_m,\quad w_k \neq \varepsilon,\ 1 \le k \le m,\ m < d$.

On an abstract level we can say that the sequence $(w_1, \ldots, w_m)$ refines the two sequences $(u_1, \ldots, u_g)$ and $(u_{g+1}, \ldots, u_d)$. Let us see what happens if we pass via the homomorphism φ to the finite semigroup S. Thus we replace the u_i and w_k by $p_i = \varphi(u_i)$ and $s_k = \varphi(w_k)$ respectively.

Two sequences $(p_1, \ldots, p_g) \in S^g$ and $(p_{g+1}, \ldots, p_d) \in S^{d-g}$ are refined to a single sequence $(s_1, \ldots, s_m) \in S^m$, $m < d$, such that each $p_i \in S$ is a product of some s_k. We shall say that $(s_1, \ldots, s_m)$ is a *common refinement* of $(p_1, \ldots, p_g)$ and $(p_{g+1}, \ldots, p_d)$.

However, for each d, there are only finitely many candidates for $(s_1, \ldots, s_m)$ with $m < d$. Hence, in a nondeterministic step, we can guess and fix such a sequence $(s_1, \ldots, s_m)$ being the φ-image of $(w_1, \ldots, w_m)$.

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

Example 12.3.1. Consider the equation $xauzau = yzbxaaby$. The solution, which was given in Example 12.1.1, 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 visualized by the following figure.

<table>
<tr><td colspan="3">a b b</td><td>a</td><td colspan="3">b a b</td><td colspan="2">b a</td><td>a</td><td colspan="3">b a b</td></tr>
<tr><td colspan="2">a b</td><td>b</td><td>a</td><td>b</td><td colspan="2">a b</td><td>b</td><td>a</td><td>a</td><td>b</td><td colspan="2">a b</td></tr>
<tr><td colspan="2">a b</td><td colspan="2">b a</td><td>b</td><td colspan="3">a b b</td><td>a</td><td>a</td><td>b</td><td colspan="2">a b</td></tr>
</table>

Passing to the semigroup $S = \{a, a^2, b, 0\}$ of Example 12.2.5, 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 $\varphi\colon A^+ \longrightarrow S$ are given as in Subsection 12.1.8. An *S-sequence* is a sequence $(s_1, \ldots, s_m) \in S^m$, $m \ge 0$. A *representation* of $(s_1, \ldots, s_m)$ is a triple $(I, \le, \varphi_I)$ such that $(I, \le)$ is a totally ordered set of

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

$m+1$ elements and

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

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

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

We have $\varphi_I(i,j) = \varepsilon$ if and only if $i = j$, and we have $\varphi_I(i,k) = \varphi_I(i,j)\varphi_I(j,k)$ 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 ρ is the identity.

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

Remark 12.3.2. 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 (cf. Lemma 12.3.6), by convention, we count only standard representations.

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 $a_{i+1}\cdots a_j$. In particular, $w = w(0,m) = w(0,i)w(i,m)$ for all $0 \leq i \leq m$. The associated S-sequence of a word w is defined by $w_S = (\varphi(a_1),\ldots,\varphi(a_m))$. The notation w_S also refers to its standard representation $w_S = (\{0,\ldots,m\},\leq,\varphi_w)$. The mapping φ_w is defined by $\varphi_w(i,j) = \varphi(w(i,j))$ for all $0 \leq i \leq j \leq m$.

Let s, s' be S-sequences, which are 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 12.3.3. 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 φ_I is the restriction of $\varphi_{I'}$ to I.

Let s be an S-sequence and $(I,\leq,\varphi_I)$ some representation. A word $w \in A^*$ is called a *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 ρ.

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., ρ becomes an inclusion and therefore $\varphi_I(i,j) = \varphi(w(i,j))$ for all $i,j \in I$, $i \leq j$.

Lemma 12.3.4. *Every S-sequence $(s_1,\dots,s_m)$ has a model $w \in A^*$.*

Proof. Since φ is surjective, there are nonempty 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 $w \models (s_1,\dots,s_m)$. ■

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 suborder 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.

Let $(I, \leq, \varphi_I)$ be a representation of some S-sequence, $T \subseteq I$ a nonempty subset, and $\ell^*, r^* \in I$ 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:

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

The intuition behind the last definition should be fairly clear. An admissible extension refines $(I, \leq, \varphi_I)$ by defining new positions between ℓ^* 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 ℓ^* 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 12.3.5. Let $(s_1,\dots,s_6)$ be some S-sequence, $(I, \leq, \varphi_I)$ its standard representation, $\ell^* = 4$ and $r^* = 6$. Let $(I^*, \leq, \varphi_{I^*})$ represent an

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

admissible extension of $(I, \leq, \varphi_I)$ by $\{0, 3, 4, 5\}$ at $[4, 6]$. Then we may assume $I^* = \{0, \ldots, 6\} \cup \{3^*, 4^*\}$. The ordering of I^* satisfies $0 < 1 < 2 < 3 < 4 < 5 < 6$ and $4 = 0^* < 3^* < 4^* < 5^* = 6$.

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 $s_6 = s_4 s_5$.

The following figure represents this admissible extension.

$$\begin{tikzcd}
0 \arrow[r, dash, "s_1"] & 1 \arrow[r, dash, "s_2"] & 2 \arrow[r, dash, "s_3"] & 3 \arrow[r, dash, "s_4"] & 4 \arrow[r, dash, "s_5"] \arrow[d, equal] & 5 \arrow[rr, dash, "s_6"] \arrow[d, equal] & & 6 \arrow[d, equal] \\
& & & & 0^* \arrow[r, dash, "s_1 s_2 s_3"] & 3^* \arrow[r, dash, "s_4"] & 4^* \arrow[r, dash, "s_5"] & 5^*
\end{tikzcd}$$

Lemma 12.3.6. *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 $\operatorname{Card} I + \operatorname{Card} T$. ∎

Example 12.3.7. Consider the same situation as in Example 12.3.5. The number of admissible extensions by the subset $\{0,\ 3,\ 4,\ 5\}$ at the interval $[4,\ 6]$ is given as a sum $e_1 + e_2 + e_3$. The numbers e_1, e_2, and e_3 respectively are the numbers of admissible extensions with $4^* \leq 5$, with $3^* < 5 < 4^*$, and with $5 \leq 3^*$ respectively. We have

$$\begin{aligned}
e_1 &= \operatorname{Card}\{s \in S^\varepsilon \mid s_5 = s_1 s_2 s_3 s_4 s,\ s_5 = s s_6\}, \\
e_2 &= \operatorname{Card}\{(r, s) \in S \times S \mid s_4 = rs,\ s_5 = s_1 s_2 s_3 r,\ s_6 = s s_5\}, \\
e_3 &= \operatorname{Card}\{s \in S^\varepsilon \mid s_1 s_2 s_3 = s_5 s,\ s_6 = s s_4 s_5\}.
\end{aligned}$$

Note that $s_1 s_2 s_3 s_4 s_5 \neq s_5 s_6$ implies $e_1 + e_2 + e_3 = 0$. Thus, there is no admissible extension of $\{0,\ 3,\ 4,\ 5\}$ at $[4,\ 6]$ in this case.

12.3.2. 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 rational constraints $L_x \subseteq A^*$ such that, without restriction, $\varepsilon \notin L_x \neq \emptyset$ for all $x \in \Omega$. Recall that we fixed a homomorphism $\varphi : A^+ \to S$ to some finite semigroup S such that $\varphi^{-1}\varphi(L_x) = L_x$ for all

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

$x \in \Omega$. 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 $x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \le g < d$, and the question is the existence of a nonsingular 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.

Let $n \ge 0$ and $\varphi : A^+ \to S$ be a homomorphism to a finite semigroup S.

(i) A *system of boundary equations* is specified by a tuple

$$\mathscr{B} = ((\Gamma, ^-), (I, \le, \varphi_I), \text{left}, B)$$

where Γ is a set of $2n$ variables, $^- : \Gamma \to \Gamma$ is an involution without fixed points, i.e., $\overline{\overline{x}} = x$, $x \ne \overline{x}$, for all $x \in \Gamma$, the triple $(I, \le, \varphi_I)$ is a linear order over S, left$: \Gamma \to I$ 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 $x \in \Gamma$, $i, j \in I$, such that left$(x) \le i$ and left$(\overline{x}) \le j$.

(ii) A *solution* of $\mathscr{B}$ is a model $w \models (I, \le, \varphi_I)$, $w \in A^*$, such that

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

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

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

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

We shall not distinguish between *isomorphic* systems. In particular, we may always think that $\Gamma = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$ and that $(I, \le, \varphi_I)$ is the standard representation of some S-sequence, $I = \{0, \ldots, m\}$ for some $n, m \ge 0$.

Remark 12.3.8. If we have $n = 0$, then there are no variables, hence no boundary equations, and any model $w \models (I, \le, \varphi_I)$ is a solution of $\mathscr{B}$. Therefore, if $n = 0$, then the system is solvable by Lemma 12.3.4.

We are now ready to pass from word equations to boundary equations. The formal description is rather technical. We will see an example later. 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

$$\mathscr{B} = ((\Gamma, ^-), (I, \le, \varphi_I), \text{left}, B)$$

of boundary equations having the following two properties.

(i) 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 $\mathscr{B}$.

(ii) Let $w \models (I,\le,\varphi_I)$ be a solution of $\mathscr{B}$. 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 $\mathscr{B}$ we start with the S-sequence

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

Let $(I,\le,\varphi_I)$ be some representation, $I = \{i_0,\ldots,i_d\}$, $i_0 \le \cdots \le i_d$. The next step is to define the pair $(\Gamma,{}^-)$ and the mapping left$:\Gamma \to I$. The intuitive meaning of $(\Gamma,{}^-)$ is that Γ is a new set of variables where the notion of *dual* is defined and that left indicates the leftmost position of a variable in a given solution. We formalize this concept by using some undirected graph. Let (V,E) be the undirected graph with vertex set $V = \{1,\ldots,d\}$ and edge set $E = \{(p,q) \in V\times V \mid x_p = x_q\}$. Clearly, each edge defines a variable, but now we have a canonical choice to define the dual of (p,q) to be (q,p).

The idea is now 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 below an extra variable x_0 (and its dual $\overline{x_0}$); in the other list of equations there is some redundancy since the edge relation in our graph is transitive. 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 do not 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 $\operatorname{Card} F = 2(d-c)$, where c is the number of connected components of (V,E). 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 $x \ne \overline{x}$ and $\overline{\overline{x}} = x$ for all $x \in F$. Taking duals corresponds to edge reversing in (V,F). Define two extra elements x_0 and $\overline{x_0}$ with $\overline{\overline{x_0}} = x_0$ and define $\Gamma = \{x_0,\overline{x_0}\} \cup F$ and

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

This defines the set Γ, the involution without fixed points $\bar{\ }:\Gamma \to \Gamma$, and the mapping left: $\Gamma \to I$. The elements of Γ are called variables again.

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

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

We still have to verify the two properties above.

(i) 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:

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

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

(ii) Let $w \models (I, \le, \varphi_I)$ be a solution of $\mathscr{B}$. 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 $(x_0, \mathrm{right}(x_0), \overline{x_0}, \mathrm{right}(\overline{x_0})) \in B$ implies $w(i_0, i_g) = w(i_g, i_d)$ and it follows that $w \in A^* vv A^*$ 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 nonempty word. The elements $(x, \mathrm{right}(x), \overline{x}, \mathrm{right}(\overline{x})) \in B$ 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 σ is well defined. We have $\varphi\sigma(x_p) = \varphi w(i_{p-1}, i_p) = \psi(x_p)$ since $w \models (I, \le, \varphi_I)$. 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 rational constraints given by the mapping ψ has a solution if and only if the system of boundary equations is solvable. The construction of the system $\mathscr{B}$ above can be performed in polynomial time (and logarithmic space). Due to this reduction, Makanin's result follows from Theorem 12.3.10. The assertion of this theorem is in fact equivalent to Makanin's result; see Lemma 12.3.12.

Example 12.3.9. We assume that the equation is simply $xyxyz = zyxyx$ and that we ignore any constraints for a moment. Hence, $\sigma(x) = a$, $\sigma(y) = b$, and $\sigma(z) = aba$, i.e., the word $v = abababa$ solves the equation. The transformation which yields the system of boundary equations is based on the following picture. The first line represents the word $w = vv$ of length 14.

<table>
<tr><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td><td>a</td><td>b</td><td>a</td><td>b</td><td>a</td><td>$\overline{b}$</td><td>$\overline{a}$</td></tr>
<tr><td colspan="7">x_0</td><td colspan="7">$\overline{x_0}$</td></tr>
<tr><td>x_1</td><td>x_2</td><td>x_3</td><td>x_4</td><td colspan="3">x_5</td><td colspan="3">$\overline{x_5}$</td><td>x_6</td><td>x_7</td><td>$\overline{x_6}$</td><td>$\overline{x_7}$</td></tr>
<tr><td></td><td></td><td>$\overline{x_1}$</td><td>$\overline{x_2}$</td><td colspan="6"></td><td>$\overline{x_4}$</td><td>$\overline{x_3}$</td><td></td><td></td></tr>
</table>

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

According to the picture above we may represent the equation by a system of word equations using a set of eight variables with their duals $\{x_0, \overline{x_0}, \ldots, x_7, \overline{x_7}\}$:

$$\begin{aligned}
x_0 &= x_1x_2x_3x_4x_5,\\
\overline{x_0} &= \overline{x_5}x_6x_7\overline{x_6}\overline{x_7},\\
\overline{x_1} &= x_3,\\
\overline{x_3} &= x_7,\\
\overline{x_2} &= x_4,\\
\overline{x_4} &= x_6,\\
x_i &= \overline{x_i} \text{ for } 0 \le i \le 7.
\end{aligned}$$

The system looks more complicated than the original equation, but the pattern is straightforward from the picture. The word vv has positions $0,\ldots,14$. We define $\mathrm{left}(x_0) = 0$, $\mathrm{left}(\overline{x_0}) = 7$, $\mathrm{left}(x_1) = 0$, $\mathrm{left}(\overline{x_1}) = 2$, $\mathrm{left}(x_2) = 1$, $\mathrm{left}(\overline{x_2}) = 3$, $\mathrm{left}(x_3) = 2$, $\mathrm{left}(\overline{x_3}) = 11$, $\mathrm{left}(x_4) = 3$, $\mathrm{left}(\overline{x_4}) = 10$, $\mathrm{left}(x_5) = 4$, $\mathrm{left}(\overline{x_5}) = 7$, $\mathrm{left}(x_6) = 10$, $\mathrm{left}(\overline{x_6}) = 12$, $\mathrm{left}(x_7) = 11$, and $\mathrm{left}(\overline{x_7}) = 13$.

The set B of boundary equations is defined by the following list:

$$\begin{gathered}
(x_0, 7, \overline{x_0}, 14), (x_1, 1, \overline{x_1}, 3), (x_2, 2, \overline{x_2}, 4), (x_3, 3, \overline{x_3}, 12),\\
(x_4, 4, \overline{x_4}, 11), (x_5, 7, \overline{x_5}, 10), (x_6, 11, \overline{x_6}, 12), (x_7, 12, \overline{x_7}, 14).
\end{gathered}$$

Since there were no constraints, the linear order is just the pair $(\{0,\ldots,14\}, \le)$.

12.3.3. The main theorem

THEOREM 12.3.10. *It is decidable whether a system of boundary equations has a solution.*

The rest of this chapter is devoted to the proof of Theorem 12.3.10. An important step is done in the next proposition: we can bound the exponent of periodicity while searching for a solution.

PROPOSITION 12.3.11. *Given as instance a system of boundary equations $\mathscr{B}$, we can compute a number $e(\mathscr{B})$ having the property that if $\mathscr{B}$ is solvable, then we have* $\exp(\mathscr{B}) \le e(\mathscr{B})$.

The proof of Proposition 12.3.11 could be based on the same techniques as presented in Section 12.2. However, for our purposes we prefer to prove Proposition 12.3.11 via a reduction to word equations.

LEMMA 12.3.12. *There is an effective reduction of the solvability of a system of boundary equations $\mathscr{B}$ to the satisfiability problem of some* *word equation with rational constraints such that for all solutions* $w \in A^*$ *of the word equation we have* $\exp(\mathscr{B}) \leq \exp(w)$.

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

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 rational constraints is solvable if and only if $\mathscr{B}$ is solvable. Indeed, if $w \in A^*$ is a solution of $\mathscr{B}$, then, by definition, we have $(I,\leq,\varphi_I) \leq_\rho w_S$, 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 $\sigma(y_p) = w(\rho(p-1),\rho(p))$, $1 \leq p \leq m$, yields a solution of the system of word equations.

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

Next, we transform the system of word equations into a single word equation $L = R$ using Proposition 12.1.8 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 $\mathscr{B}$. Hence $\exp(\mathscr{B}) \leq \exp(v) \leq \exp(w)$. This yields Lemma 12.3.12. Now, let d be the denotational length of $Ly_1\cdots y_m = Ry_1\cdots y_m$. Then define the number $e(\mathscr{B}) = e(c(S),d)$, which has been given in Theorem 12.2.1. We can choose w such that $\exp(w) \leq e(c(S),d)$. This proves Proposition 12.3.11. ■

12.3.4. The convex chain condition

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

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

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

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

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 $x \in \Gamma$. Clearly, this is no restriction. For $x \in \Gamma$ 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 nonempty subset $\{i_1, \ldots, i_r\} \subseteq I$ such that $\operatorname{left}(x) \leq i_1 \leq \cdots \leq i_r$. The *right boundary* of x is defined by $\operatorname{right}(x) = i_r$.

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

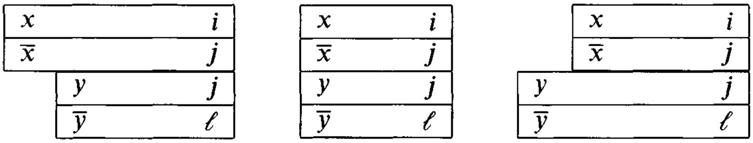
Let $(x, i, \overline{x}, j), (y, i, \overline{y}, j), (y, i', \overline{y}, j') \in B$. Then we assume from now on

- $\operatorname{left}(x) \leq \operatorname{left}(\overline{x})$ if and only if $i \leq j$,
- $\varphi_I(\operatorname{left}(x), i) = \varphi_I(\operatorname{left}(\overline{x}), j)$,
- $\operatorname{left}(x) \leq \operatorname{left}(y)$ if and only if $\operatorname{left}(\overline{x}) \leq \operatorname{left}(\overline{y})$,
- $i \leq i'$ 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 $\operatorname{left}(x) \leq i_1 \leq \cdots \leq i_r$, then we also have $\operatorname{left}(\overline{x}) \leq j_1 \leq \cdots \leq j_r$. In particular, $B(x)$ contains a brick $(x, \operatorname{right}(x), \overline{x}, \operatorname{right}(\overline{x}))$. The set $B(x)$ can be depicted as follows:

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

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



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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

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

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

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

For a chain C and a variable $x \in \Gamma$ we define the x-length $|C|_x$ of C to be the number of bricks in C having label x. Thus, the length of a chain C is the sum $\sum\limits_{x \in \Gamma} |C|_x$.

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

$$\begin{aligned} \mathrm{left}(\overline{x_p}) &\geq \mathrm{left}(x_{p+1}) \text{ for } 1 \leq p < q, \\ \mathrm{left}(\overline{x_p}) &\leq \mathrm{left}(x_{p+1}) \text{ for } q \leq p < k. \end{aligned}$$

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* to a brick $(x', i', \overline{x}', j')$, if there is a convex chain C of length k as above for some $k \geq 1$ such that $(x, i, \overline{x}, j) = (x_1, i_1, \overline{x_1}, i_2)$, and $(x', i', \overline{x}', j') = (x_k, i_k, \overline{x_k}, i_{k+1})$.

Remark 12.3.13. If $C = (b_1, \dots, b_k)$ is a convex chain, then its dual $\overline{C} = (\overline{b_k}, \dots, \overline{b_1})$ and $(b_p, \dots, b_q)$, $1 \leq p \leq q \leq k$ are convex chains. If $b_p = (x_p, i_p, \overline{x_p}, i_p)$ for some $1 < p < k$, then $(b_1, \dots, b_{p-1}, b_{p+1}, \dots b_k)$ is a convex chain. If $b_p = b_q$ for some $1 < p < q \leq k$, then $(b_1, \dots, b_{p-1}, b_q, \dots b_k)$ 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 always clean.

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 $\mathscr{B}$ 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*.

In the following we concentrate on solvable systems and we need a few more notations. Let $\mathscr{B} = ((\Gamma, \bar{\ }), (I, \leq, \varphi_I), \mathrm{left}, B)$ be a solvable system of boundary equations and $w \in \Gamma^*$ such that $w \models (I, \leq, \varphi_I)$ is a solution of $\mathscr{B}$. Since w is a solution we may assume that I is a subset of positions of w. For all $x \in \Gamma$ define a word $w(x) \in A^*$ by

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

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

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

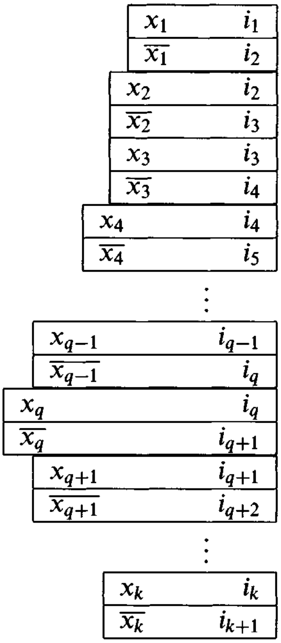


Figure 12.2. A convex chain.

Moreover, for each brick $b = (x, i, \overline{x}, j) \in B$ we also define its w-length by

$$|b|_w = |w(\text{left}(x), i)|.$$

For all $x \in \Gamma$ and $b \in B$ we have $w(x) = w(\overline{x})$, $|x|_w = |\overline{x}|_w$, $|b|_w = |\overline{b}|_w$, and $|b|_w \leq |x|_w$, if x is the label of b. A brick is uniquely determined by its label and its w-length $|b|_w$. A singly exponential bound on the number of bricks as given in the next lemma is due to Gutiérrez 1998a. The improvement on this number has been essential in order to obtain the singly exponential complexity bound in Theorem 12.4.2 below.

Lemma 12.3.14. *Let $n, m, f \in \mathbb{N}$ and $\mathcal{B} = ((\Gamma, ^{-}), (I, \leq, \varphi_I), \text{left}, B)$ be a solvable system of boundary equations such that $w \models (I, \leq, \varphi_I)$ is a solution of $\mathcal{B}$. Let* $\operatorname{Card}\Gamma = 2n$, *$F \subseteq I$, and* $\operatorname{Card} F = f$. *Suppose that every brick $b \in B$ can be linked via a convex chain C to a basis with respect to F such that for each $x \in \Gamma$ the number of bricks in C having label x is at most m, i.e., $|C|_x \leq m$.*

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Then we can bound the size of B by

$$\operatorname{Card} B \leq 2n \cdot f \cdot (2m+1)^n.$$

Proof. Consider a convex chain C of length k such that $|C|_x \leq m$ for all $x \in \Gamma$ and where the last brick is a basis:

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

There are $2n$ possibilities for the label of the first brick. We shall calculate an upper bound for the number of possible w-lengths for the first brick $(x_1, i_1, \overline{x_1}, i_2)$. The length of the first brick is determined by the w-length of the last brick $(x_k, i_k, \overline{x_k}, i_{k+1})$ and by summing up the values $\operatorname{left}(x_{i+1}) - \operatorname{left}(\overline{x_i})$ for $i = k-1, \ldots, 1$; see Figure 12.2. Recall that $i \in I$ denotes a position in the solution w, hence $\operatorname{left}(x_{i+1}) - \operatorname{left}(\overline{x_i}) \in \mathbb{Z}$. So the w-length of the first brick is

$$i_{k+1} - \operatorname{left}(\overline{x_k}) + \operatorname{left}(x_k) - \operatorname{left}(\overline{x_{k-1}}) + \cdots + \operatorname{left}(x_2) - \operatorname{left}(\overline{x_1}).$$

Then we can rearrange this sum in some formula of type

$$i_{k+1} - \operatorname{left}(\overline{x_1}) + \sum_{x \in \Gamma} m_x \cdot \big(\operatorname{left}(x) - \operatorname{left}(\overline{x})\big)$$

where due to the hypothesis on C we have $-m \leq m_x \leq m$. The value $\operatorname{left}(\overline{x_1})$ is uniquely determined by the label x_1 and i_{k+1} is a basis. Hence at most $f \cdot (2m+1)^n$ different values can be produced using these sums, when the label x_1 is fixed. Thus, at most

$$2n \cdot f \cdot (2m+1)^n.$$

different first bricks are possible. But this is also an upper bound for the number of bricks $\operatorname{Card} B$ by the convex chain condition. ∎

Every system of boundary equations $\mathscr{B}$ satisfies the convex chain condition with respect to the set I, trivially. Furthermore, if we construct $\mathscr{B}$ by starting from a word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \leq g < d$, then we have $\operatorname{Card} I \leq d$. The transformation rules below will increase neither the number $2n$ of variables nor the sum $2n + f$. They will increase the sizes of I and of B. However, Lemma 12.3.14 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 (or skyscrapers) lead to high domino towers and hence to a lower bound on the exponent of periodicity in any solution.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

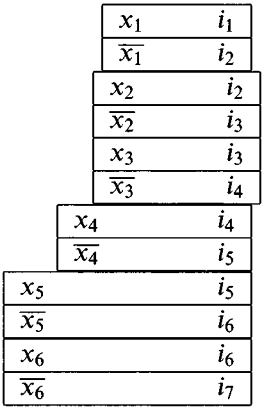


Figure 12.3. The upper part of a convex chain.

Proposition 12.3.15. *Let $n, m \in \mathbb{N}$ and $\mathscr{B} = ((\Gamma, ^-), (I, \leq, \varphi_I), \mathrm{left}, B)$ be a solvable system of boundary equations with* $\mathrm{Card}\, \Gamma = 2n$. *Let $w \models (I, \leq, \varphi_I)$ be a solution of $\mathscr{B}$. Suppose that there is at least one clean convex chain such that $m \leq |C|_x$ for some $x \in \Gamma$. Then we have the following lower bound for the exponent of periodicity of the solution w:*

$$m \leq 2n \cdot (\exp(w) + 1) - 1.$$

Proof. The hypothesis implies $n \neq 0$, hence $w \neq \varepsilon$. The assertion is trivial for $m < 4n$. Hence let $n \geq 1$ and $m \geq 4n$. Define $h = \left\lceil \frac{m+1}{2n} \right\rceil$. We have $h \geq 3$. (Eventually h will be the height of some domino tower.)

Let $C = (b_1, \ldots, b_k)$ be a clean convex chain such that $m \leq |C|_x$ for some $x \in \Gamma$. Let $b_p = (x_{i_p}, i_p, \overline{x_{i_p}}, i_{p+1})$ for $1 \leq p \leq k$. Define $m' = \left\lceil \frac{m+1}{2} \right\rceil$; then by duality (replacing C by $\overline{C}$ and x by $\overline{x}$) we may assume that the label x occurs at least m' times in the upper part up to some k' where $k' \leq k$ such that

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

This upper part of the chain C up to k' might look as in Figure 12.3.

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

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

such that $n' \leq n$ and for all q, j where $p_{j-1} < q \leq p_j$, $1 \leq j \leq n'$, 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 $j \in \{1, \ldots, n'\}$ such that the number of occurrences of the label x in the interval $[p_{j-1} + 1, p_j]$ is at least

$$\left\lceil \frac{m+1}{2n} \right\rceil.$$

We conclude that (after renaming) there are a clean convex chain $C = (b_1, \ldots, b_\ell)$ and a variable $x \in \Gamma$ having the following properties:

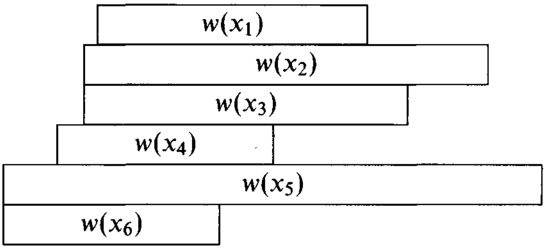
$$\begin{array}{rcllc}
|C|_x & = & \left\lceil \frac{m+1}{2n} \right\rceil, & & \\
\mathrm{left}(\overline{x_p}) & \geq & \mathrm{left}(x_{p+1}) & \text{for} & 1 \leq p < \ell, \\
|x_p|_w & \geq & |x_\ell|_w & \text{for} & 1 \leq p \leq \ell.
\end{array}$$

Recall that $h = \left\lceil \frac{m+1}{2n} \right\rceil$. We have $h \geq 3$ and the label x occurs exactly h times in the clean convex chain C. By cutting off the sequence we may assume that x is the first label x_1.

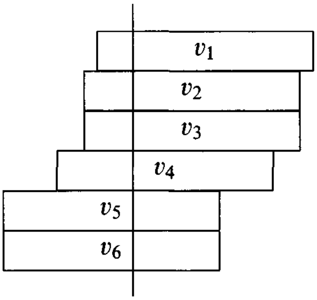
This is the point where we switch from the chain to the sequence of words

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

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 $u_{q_i} \neq u_{q_j}$ for all $1 \leq i, j \leq h$, $i \neq j$. (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 Lemma 12.1.4 we obtain $h - 1 \leq \exp(w_1) \leq \exp(w)$. The assertion of the proposition follows. ∎

Corollary 12.3.16. *Let $\mathscr{B} = ((\Gamma, \bar{\ }), (I, \leq, \varphi_I), \mathrm{left}, B)$ denote a solvable system of boundary equations which satisfies the convex chain condition with respect to some subset $F \subseteq I$. Let $\mathrm{Card}\,\Gamma = 2n$ and $\mathrm{Card}\,F = f$. Then we have*

$$\mathrm{Card}\,B \leq 2n \cdot f \cdot (4n \cdot (\exp(\mathscr{B}) + 1))^n.$$

If moreover $\mathrm{Card}\,\Gamma, \mathrm{Card}\,F \in O(d)$, and $\exp(\mathscr{B}) \in 2^{O(d+\log c(S))}$, then we have

$$\mathrm{Card}\,B \in 2^{O(d^2+d\log c(S))}.$$

Proof. Let $2n = \mathrm{Card}\,\Gamma$, $f = \mathrm{Card}\,F$, and m be the maximal x-length of a clean convex chain, $x \in \Gamma$. By Remark 12.3.13 and Lemma 12.3.14 we have

$$\mathrm{Card}\,B \leq 2n \cdot f \cdot (2m+1)^n.$$

Choose a solution w such that $\exp(w) \leq \exp(\mathscr{B})$. Proposition 12.3.15 yields

$$m \leq 2n \cdot (\exp(w) + 1) - 1.$$

Putting things together we obtain

$$\operatorname{Card} B \leq 2n \cdot f \cdot (4n \cdot (\exp(w) + 1))^n \leq 2n \cdot f \cdot (4n \cdot (\exp(\mathscr{B}) + 1))^n.$$

The result follows. ∎

12.3.5. Transformation rules

We are ready to define the (nondeterministic) transformation rules of Makanin's algorithm. If we apply a rule to a system $\mathscr{B} = ((\Gamma, {}^-), (I, \leq, \varphi_I), \text{left}, B)$, then the new system is denoted by $\mathscr{B}' = ((\Gamma', {}^-), (I', \leq, \varphi_{I'}), \text{left}', B')$. The transformation rules below will have the property that if $\mathscr{B} = ((\Gamma, {}^-), (I, \leq, \varphi_I), \text{left}, B)$ satisfies the convex chain condition with respect to some subset $F \subseteq I$, then $\mathscr{B}'$ satisfies the convex chain condition with respect to some subset $F' \subseteq I'$ such that $\operatorname{Card} \Gamma' + \operatorname{Card} F' \leq \operatorname{Card} \Gamma + \operatorname{Card} F$. Thus, if we start with a system $\mathscr{B}_0$ where $\operatorname{Card} \Gamma_0 = 2n_0$ and $\operatorname{Card} I_0 \leq d$, 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 (nondeterministic) rule is *downward correct*, provided the following condition holds: if $w \in A^*$ is a solution of $\mathscr{B}$, then (for at least one nondeterministic choice) some suffix w' of w is a solution of $\mathscr{B}'$, and moreover either $\operatorname{Card} \Gamma' < \operatorname{Card} \Gamma$ 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 (nondeterministic) rule is *upward correct*, provided the following condition holds: if $w' \in A^*$ is a solution of $\mathscr{B}'$ (and $\mathscr{B}'$ is the result of any nondeterministic choice), then there is word $w \in A^*$, which is a solution of $\mathscr{B}$.

Rule 1. If there is some $x \in \Gamma$ with $\text{left}(x) = \text{right}(x)$, 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 Γ.

Remark 12.3.17. Obviously Rule 1 is upward and downward correct since we have $w(i,i) = \varepsilon$ 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 $x \in \Gamma$ be such that $\text{left}(x) = \text{right}(x)$ and assume that $x, \overline{x}$ are canceled by Rule 1. Define $F' = F \cup \{\text{left}(x), \text{left}(\overline{x})\}$. 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 $\mathrm{right}(x) = \mathrm{left}(x) \in F'$. Thus, if $\mathscr{B}$ satisfies the convex chain condition with respect to F, then the system $\mathscr{B}'$ (after an application of Rule 1) satisfies the convex chain condition with respect to F'. We have $\mathrm{Card}\,\Gamma' + \mathrm{Card}\,F' \leq \mathrm{Card}\,\Gamma + \mathrm{Card}\,F$.

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

Remark 12.3.18. Recall that for $(x, i, \overline{x}, j) \in B$ we have $\mathrm{left}(x) = \mathrm{left}(\overline{x})$ if and only if $i = j$. Thus, if $\mathrm{left}(x) = \mathrm{left}(\overline{x})$, 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 $\ell \notin \mathrm{left}(\Gamma)$, then cancel the index ℓ from I. This means we replace the linear order over S by the induced suborder $(I', \leq, \varphi_{I'})$ where $I' = I \setminus \{\ell\}$.

Remark 12.3.19. 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 nonempty 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 $\mathscr{B}$. 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 $\mathscr{B} = ((\Gamma, ^-), (I, \leq, \varphi_I), \mathrm{left}, B)$, we apply Rules 1, 2 and 3 as often as possible. In particular, we shall assume that $\mathrm{left}(x) < \mathrm{right}(x)$, $\mathrm{left}(x) \neq \mathrm{left}(\overline{x})$ for all $x \in \Gamma$, and that there exists some $x \in \Gamma$ with $\mathrm{left}(x) = \min(I)$.

Rule 4. We divide Rule 4 into six steps.

We need some notation. Define $\ell = \min(I)$ and $r = \max\{\mathrm{right}(x) \mid x \in \Gamma,\ \mathrm{left}(x) = \ell\}$. Note that $\ell \in \mathrm{left}(\Gamma)$, hence $r \in I$ exists and we have $\ell < r$. Choose (and fix) some $x_o \in \Gamma$ with $\mathrm{left}(x_o) = \ell$ and $\mathrm{right}(x_o) = r$.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Define $\ell^* = \mathrm{left}(\overline{x_o})$ and $r^* = \mathrm{right}(\overline{x_o})$. Define the *critical boundary* $c \in I$ by $c = \min\{c', r\}$ where

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

Note that since $r < r^* = \mathrm{right}(\overline{x_o})$, the minimum c' and hence the critical boundary c exist. We have $\ell < c \leq r < r^*$ and $c \leq \ell^* < r^*$. The ordering of r and ℓ^* depends on the system; it is of no importance.

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

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

Note that $\min(T) = \ell$ and that $i \in T$ for all $(x_o, i, \overline{x_o}, j) \in B$. Moreover, since $\mathrm{left}(x) < c$ implies $\mathrm{right}(x) \leq r$, 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 $\min(T^*) = \ell^*$ and $\max(T^*) = r^*$ and such that T^* is in order-respecting bijection with T. For each $i \in T$ the corresponding position in T^* is denoted by 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, for all $(x, i, \overline{x}, j) \in B$ with $\mathrm{left}(x) < c$ we require

$$\begin{aligned} \mathrm{left}(x)^* = \mathrm{left}(\overline{x}) \quad &\Leftrightarrow \quad i^* = j, \\ \mathrm{left}(x)^* < \mathrm{left}(\overline{x}) \quad &\Leftrightarrow \quad i^* < j. \end{aligned}$$

In particular, for all bricks $(x_o, i, \overline{x_o}, j)$ we require $i^* = j$. If such an admissible extension is not possible, then Step 1 cannot be completed and Rule 4 is not applicable.

Step 2. Introduce a new variable x_v and its dual $\overline{x_v}$. We define $\mathrm{left}(x_v) = c$, $\mathrm{left}(\overline{x_v}) = c^*$. For all $i \in T$ such that there is some $(x, i, \overline{x}, j) \in B$ with $\mathrm{left}(x) < c \leq i$ introduce new bricks $(x_v, i, \overline{x_v}, i^*)$ and $(\overline{x_v}, i^*, x_v, i)$.

Step 3. As long as there is a variable $x \in \Gamma$ with $\mathrm{left}(x) < c$, replace $\mathrm{left}(x)$ by $\mathrm{left}'(x) = \mathrm{left}(x)^*$ 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 12.3.20. 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 $\text{left}(x) = \text{left}'(x')$, then sometimes we may still write $x = x'$. In particular, $x_v = x'_v$, $\overline{x_v} = \overline{x_v}'$, $\overline{x_o} = \overline{x_o}'$, but $x_o \neq x_o'$.

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

$$\begin{array}{lll} b' = (x', i^*, \overline{x}', j^*) & \text{if } \text{left}(x) < c, & \text{left}(\overline{x}) < c, \\ b' = (x', i^*, \overline{x}, j) & \text{if } \text{left}(x) < c, & c \leq \text{left}(\overline{x}), \\ b' = (x, i, \overline{x}', j^*) & \text{if } c \leq \text{left}(x), & \text{left}(\overline{x}) < c, \\ b' = (x, i, \overline{x}, j) & \text{if } c \leq \text{left}(x), & c \leq \text{left}(\overline{x}). \end{array}$$

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 $\mathscr{B}' = ((\Gamma', ^-), (I', \leq, \varphi_{I'}), \text{left}', B')$. We will show from Lemmas 12.3.25 to 12.3.28 that $\mathscr{B}'$ satisfies the convex chain condition with respect to F'. The first lemma is a trivial observation.

Lemma 12.3.21. *We have* $\operatorname{Card}\Gamma' = \operatorname{Card}\Gamma$ *and* $\operatorname{Card}F' \leq \operatorname{Card}F$.

Proof. In Step 2 new variables x_v and $\overline{x_v}$ are introduced, but in Step 5 the variables x'_o and $\overline{x_o}$ are canceled. Hence $\operatorname{Card}\Gamma' = \operatorname{Card}\Gamma$. The set of final indices is changed in Step 4 in such a way that $\operatorname{Card}F' \leq \operatorname{Card}F$. ■

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

Lemma 12.3.22. *Let* $\beta' = \operatorname{Card}\{(x', i', \overline{x}', j') \in B' \mid \text{left}'(x') < i'\}$ *and* $\beta = \operatorname{Card}\{(x, i, \overline{x}, j) \in B \mid \text{left}(x) < i\}$. *Then we have*

$$2\operatorname{Card}I' - \beta' \leq 2\operatorname{Card}I - \beta.$$

Proof. The inequality can be destroyed either by a new position $i^* \in T^* \backslash I$ or by the cancellation 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 β and β' and that $\text{left}(x_o) = \ell$,

$\text{left}'(x'_o) = \ell^*$.) The cancellation 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 $(x_v, i, \overline{x_v}, i^*)$, $(\overline{x_v}, i^*, x_v, i) \in B'$ and we have $\text{left}(x_v) < i$, $\text{left}(\overline{x_v}) < i^*$. Next consider $i^* = c^*$. At least one position (namely ℓ) 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 12.3.23. *Rule 4 is downward correct.*

Proof. Let $w \in A^*$ be a solution of $\mathscr{B}$. 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 nonempty 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 $\mathscr{B}'$. ■

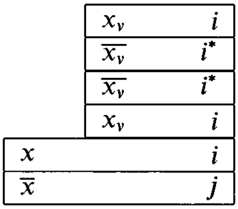
Lemma 12.3.24. *Rule 4 is upward correct.*

Proof. Let $w' \in A^*$ be a solution of $\mathscr{B}'$. 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 $(x_v, i, \overline{x_v}, i^*)$ 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 $\mathscr{B}$. ■

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

Lemma 12.3.25. *Step 2 preserves the convex chain condition with respect to the set F.*

Proof. The new bricks in Step 2 have the forms $(x_v, i, \overline{x_v}, i^*)$ and $(\overline{x_v}, i^*, x_v, i)$ for some $(x, i, \overline{x}, j) \in B$ with $\mathrm{left}(x) < c = \mathrm{left}(x_v) \leq i$. 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:



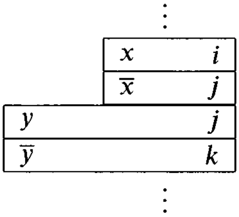
■

Lemma 12.3.26. *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.*

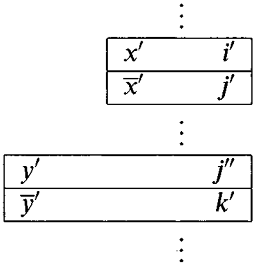
Proof. Let us have a local look at the convex chain

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

By symmetry we may assume that $\mathrm{left}(\overline{x}) \geq \mathrm{left}(y)$. 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.



https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

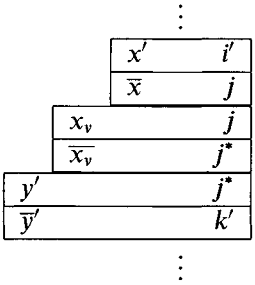
The question is whether or not $j' = j''$. If $j' = j^*$ or $j'' = j$, then we have $j' = j''$, and the chain is not broken. Hence we have to consider the case $j' = j$ and $j'' = j^*$, only. This case is equivalent to

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

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

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

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 forms $(x'_o, i^*, \overline{x_o}, i^*)$ 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 12.3.27. *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 $j \in F$. Lemma 12.3.26 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 also solves 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 $(\overline{x_v}, j^*, x_v, j)$ and we have $\text{left}'(\overline{x}') < c^* = \text{left}(\overline{x_v})$. We may put the brick $(x', i', \overline{x}', j^*)$ upon the basis $(\overline{x_v}, j^*, x_v, j)$. 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^*
	$\overline{x_v}$	j^*
	x_v	j

■

Lemma 12.3.28. *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 lemmas above yield the following proposition

Proposition 12.3.29. *Rule 4 is upward and downward correct. It preserves the convex chain condition.*

Example 12.3.30. Let $x_1 \cdots x_g = x_{g+1} \cdots x_d$ be a word equation, $1 \le g < d$, such that the rational constraints are given by a mapping $\psi : \Omega \to S$. Let

$$\mathscr{B} = ((\Gamma, ^{-}), (I, \le, \varphi_I), \text{left}, B)$$

be the result of the (log-space) reduction presented in Subsection 12.3.2. Recall that $(I, \le, \varphi_I)$ represents the S-sequence

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

We may assume that $(I, \le, \varphi_I)$ is in its standard representation, $I = \{0, \ldots, d\}$. According to the reduction the set Γ contains two variables x_0 and $\overline{x_0}$ such that $\text{left}(x_0) = 0$, $\text{right}(x_0) = g = \text{left}(\overline{x_0})$, and $\text{right}(\overline{x_0}) = d$. 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 $\text{Card}\, I = d + 1$ and $\text{Card}\, \Gamma = \text{Card}\, B \le 2d$. If the word equation has a nonsingular solution satisfying the rational constraints, then $\exp(\mathscr{B}) \le 2 \cdot e(c(S), d)$.

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

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

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 nonsingular 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 rational constraints given by ψ. With the help of this word Step 1 can be completed: Define $w = vv$; then we have

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

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

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

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 $\operatorname{Card} I^* \leq d + g - 1$. After renaming we may assume $I^* = \{0, \dots, d\} \cup T^*$ and $T^* = \{0^*, \dots, g^*\}$ where $0^* = c = g$ and $c^* = g^* = d$. This completes Step 1 of Rule 4. Since in reality we usually do not know v, the choice of I^* is a nondeterministic guess!

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

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 $\mathscr{B}'$ denote the new system; then the word v is a solution, $v \models (I', \leq, \varphi_(I')$.

Since in the present situation $\operatorname{left}(x_v) = \operatorname{right}(x_v) = g$, Rule 1 is now applicable to $\mathscr{B}'$, it cancels the superfluous bricks $(x_v, g, \overline{x_v}, d)$, $(\overline{x_v}, d, x_v, g)$ and the variables x_v and $\overline{x_v}$. The new system after an application of Rule 1 is denoted by $\mathscr{B}'' = ((\Gamma_0'', ^-), (I_0'', \leq, \varphi_{I_0''}), \operatorname{left}_0'', B_0'')$. We have $\operatorname{Card} I'' \leq d$, $\operatorname{Card} \Gamma'' = \operatorname{Card} B'' \leq 2(d-1)$. It is now the word v which is a solution of $\mathscr{B}''$, hence $\exp(\mathscr{B}'') \leq \exp(v)$. Therefore we can choose $e(\mathscr{B}'') = e(c(S), d)$.

12.4. Proof of Theorem 12.3.10

12.4.1. Decidability

The proof of Theorem 12.3.10 is now a reduction to a reachability problem in some finite directed graph.

The instance is a system of boundary equations

$$\mathscr{B}_0 = ((\Gamma_0, {}^-), (I_0, \le, \varphi_{I_0}), \mathrm{left}_0, B_0).$$

We may assume that $\mathscr{B}_0$ satisfies the assumptions made at the beginning of Subsection 12.3.4, because otherwise $\mathscr{B}_0$ is not solvable. For trivial reasons the system $\mathscr{B}_0$ satisfies the convex chain condition with respect to the set $F_0 = I_0$.

Let $2n_0 = \operatorname{Card}\Gamma_0$ and $f_0 = \operatorname{Card} F_0 = \operatorname{Card} I_0$. In accordance with Proposition 12.3.11 choose a number $e(\mathscr{B}_0)$ such that either $\mathscr{B}_0$ is not solvable or $\exp(w) \le e(\mathscr{B}_0)$ for some solution w of $\mathscr{B}_0$. Define an integer $\beta_{\max}$ by

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

Note that this value is defined just to fit Corollary 12.3.16 for a set of final indices having size at most $2n_0 + f_0$.

Now, consider a directed graph $\mathscr{G}$ (the search graph of Makanin's algorithm), which is defined as follows. The nodes of $\mathscr{G}$ are the systems of boundary equations $\mathscr{B} = ((\Gamma, {}^-), (I, \le, \varphi_I), \mathrm{left}, B)$, where

$$\begin{aligned} \operatorname{Card}\Gamma &\le 2n_0, \\ \operatorname{Card} I &\le \frac{n_0 + 2}{2} \cdot \beta_{\max}, \\ \operatorname{Card} B &\le \beta_{\max}. \end{aligned}$$

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

Clearly, $\mathscr{B}_0 \in \mathscr{G}$ and the search graph $\mathscr{G}$ has only finitely many nodes. Hence, it is enough to show the following claim: the system $\mathscr{B}_0$ has a solution if and only if there is a directed path in $\mathscr{G}$ from $\mathscr{B}_0$ 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 Lemma 12.3.4. For the "only if"-direction let $\mathscr{B}_0$ be solvable and let $w_0 \models \left(I_0, \le, \varphi_{I_0}\right)$ be a solution satisfying $\exp(w_0) \le \exp(\mathscr{B}_0)$.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

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

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

If $\mathscr{B}_M$ is a system of boundary equations without variables, then we stop. Otherwise, since $\mathscr{B}_M$ is solvable, a transformation rule is applicable. Consequently, the sequence can be continued by some solvable system $\mathscr{B}_{M+1}$ satisfying all the 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 on reaching $\mathscr{B}_M$. Let us show that all $\mathscr{B}_k$ are nodes of $\mathscr{G}$ for all $0 \leq k \leq M$. This will imply the claim since then there is a directed path to $\mathscr{B}_M$, and $\mathscr{B}_M$ is a terminal node.

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

The assertion $\mathrm{Card}\,\Gamma_k \leq 2n_0$ is trivial. The second property of the sequence implies $\exp(\mathscr{B}_k) \leq \exp(w_k) \leq \exp(w_0) \leq e(\mathscr{B}_0)$. By Corollary 12.3.16 and the fourth property we have $\mathrm{Card}\,B_k \leq \beta_{\max}$. The next lemma yields an invariant which will give the desired bound on the size of every I_k.

Lemma 12.4.1. *For* $0 \leq k \leq M$ *define* $\beta_k = \mathrm{Card}\{(x, i, \overline{x}, j) \in B_k \mid \mathrm{left}_k(x) < i\}$. *Then for all* $1 \leq k \leq M$ *we have*

$$2\,\mathrm{Card}\,I_k - \beta_k + \frac{\mathrm{Card}\,\Gamma_k}{2} \cdot \beta_{\max} \leq 2\,\mathrm{Card}\,I_{k-1} - \beta_{k-1} + \frac{\mathrm{Card}\,\Gamma_{k-1}}{2} \cdot \beta_{\max}.$$

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

$$\begin{aligned} \mathrm{Card}\,\Gamma_k &= \mathrm{Card}\,\Gamma_{k-1} - 2, \\ \mathrm{Card}\,I_k &= \mathrm{Card}\,I_{k-1}, \\ \beta_{k-1} - \beta_k &\leq \beta_{\max}. \end{aligned}$$

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

For Rule 3 we have

$$\begin{aligned}\operatorname{Card}\Gamma_k &= \operatorname{Card}\Gamma_{k-1},\\ \operatorname{Card} I_k &= \operatorname{Card} I_{k-1} - 1,\\ \beta_k &= \beta_{k-1}.\end{aligned}$$

Finally, for Rule 4 we have $\operatorname{Card}\Gamma_k = \operatorname{Card}\Gamma_{k-1}$ and Lemma 12.3.22 says

$$2\operatorname{Card} I_k - \beta_k \leq 2\operatorname{Card} I_{k-1} - \beta_{k-1}.$$

The assertion of the lemma follows. ∎

A consequence of Lemma 12.4.1 (and $\beta_k \leq \beta_{\max}$) is

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

Since $\operatorname{Card} I_0 \leq \frac{1}{2}\beta_{\max}$, we obtain $\operatorname{Card} I_k \leq \frac{n_0+2}{2}\beta_{\max}$. Hence $\mathscr{B}_k \in \mathscr{G}$ for all $0 \leq k \leq M$. This proves Theorem 12.3.10, hence Makanin's result.

12.4.2. The complexity of Makanin's algorithm

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

A node $\mathscr{B} = ((\Gamma, ^-), (I, \leq, \varphi_I), \text{left}, B)$ of the search graph $\mathscr{G}$ is encoded as a binary string over $\{0,1\}$ as follows. The code for $(\Gamma, ^-)$ is simply the number n written in binary such that $|\Gamma| = 2n$. Thus, $O(\log n_0)$ bits are enough for this part. The linear order $(I, \leq, \varphi_I)$ is encoded by its underlying S-sequence. For this part $O(n_0\beta_{\max} \log |S|)$ bits are used. The mapping $\text{left}: \Gamma \to I$ is encoded by using $O(n_0 \log(n_0\beta_{\max}))$ bits. Finally, the set of bricks B can be encoded by using $O(\beta_{\max} \log(n_0\beta_{\max}))$ bits. Note that $n_0 \leq \log \beta_{\max}$. It follows that there is effectively a constant $c \in \mathbb{N}$ such that every $\mathscr{B} \in \mathscr{G}$ 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 nondeterministic procedure. It is at most exponential in the input size.

Consider the original question whether a given word equation $x_1 \cdots x_g = x_{g+1} \cdots x_d$, $1 \leq g < d$, with rational constraints has a solution. We may assume that each rational language $L_x \subseteq A^*$ is specified by a nondeterministic finite automaton 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 Remark 12.1.12 we may assume

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

that S satisfies $\operatorname{Card} S \leq 2^{r^2}$ and $c(S) \leq r!$ By Theorem 12.2.1 choose a value $e(c(S), d) \in c(S) \cdot 2^{O(d)} \subseteq 2^{O(d + r \log r)}$ such that $e(c(S), d)$ is an upper bound for the exponent of periodicity. Transform the word equation (by a nondeterministic guess) into a system of boundary equations

$$\mathscr{B}_0 = ((\Gamma_0, {}^{-}), (I_0, \leq, \varphi_{I_0}), \mathrm{left}_0, B_0)$$

such that the word equation has a solution satisfying the rational constraints if and only if $\mathscr{B}_0$ is solvable. This is possible in such a way that, first, $\operatorname{Card} I_0, \operatorname{Card} \Gamma_0, \operatorname{Card} B_0 \in O(d)$, and, second, if $\mathscr{B}_0$ is solvable, then

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

More precisely, by Example 12.3.30 we can say $\operatorname{Card} I_0 \leq d - 1$, $\operatorname{Card} \Gamma_0 = \operatorname{Card} B_0 \leq 2(d-1)$ and, if $\mathscr{B}_0$ is solvable, then $e(\mathscr{B}_0) \leq e(c(S), d)$.

Compute a value $\beta_{\max} \in 2^{\theta\left(d^2 + dr \log r\right)}$ such that $\beta_{\max}$ is an upper bound for the number of boundary equations of each node in the search graph $\mathscr{G}$. The value $\beta_{\max}$ can be taken large enough to perform all computations over the semigroup S and it can be taken small enough in order to solve the reachability problem in the search graph $\mathscr{G}$ in nondeterministic space $\mathrm{NSPACE}(2^{O\left(d^2 + dr \log r\right)})$. By Savitch's theorem (see e.g. Hopcroft and Ullman 1979) this is equal to $\mathrm{DSPACE}\left(2^{O\left(d^2 + dr \log r\right)}\right)$. Hence we can state the final result of this chapter.

Theorem 12.4.2. *The space requirement of Makanin's algorithm for word equations with rational constraints is at most exponential space. More precisely, we have the following complexity bound:*

$$\mathrm{DSPACE}\left(2^{O\left(d^2 + dr \log r\right)}\right).$$

Remark 12.4.3. The theorem above is an assertion on Makanin's algorithm and therefore it is no statement about the inherent complexity of the satisfiability problem for word equations. In fact, by Plandowski 1999b we know that the satisfiability problem for word equations can be solved in polynomial space; see the Notes below.

Problems

Section 12.1

12.1.1 Decide whether the solution *abbababbaabab* given in Example 12.1.1 is a nonsingular solution of minimal length.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

12.1.2 Let $\Omega = \{x, y\}$ and $u, v \in A^*$ be words. Give necessary and sufficient conditions on u and v such that the equation $xu = vy$ is solvable.

12.1.3 Reduce the satisfiability problem of word equations to the satisfiability problem of systems of word equations where each variable occurs at most three times.

12.1.4 (Thierry Arnoux) Let $n > 0$. Consider the following word equation with rational constraints:

$$A = \{a, b\}, \quad \Omega = \{x_i \mid 0 \le i \le n\},$$
$$L_{x_0} = A^*, \quad L_{x_i} = aA^* \backslash (A^* b^i A^*) \text{ for } i > 0,$$
$$x_n ab^n x_n = a\ abx_0ax_0\ ab^2x_1abx_1\ \cdots\ ab^nx_{n-1}ab^{n-1}x_{n-1}\,.$$

The denotational length of this equation is $d = n^2 + 5n + 4$. Show that there is only one solution satisfying the rational constraints, and that the length grows exponentially in n.

12.1.5 Show that the solvability of word equations becomes undecidable, if the constraints are allowed to be deterministic context-free languages.
(*Hint*: It is well known that the emptiness problem for intersections of deterministic context-free languages is undecidable.)

Section 12.2

12.2.1 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.

12.2.2 Prove Propositions 12.1.7, 12.1.8, and 12.1.9. Show that the results remain true when there are rational constraints.

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

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Section 12.3

12.3.1 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 of the coefficients in each row of D is in $O(1)$ and $\|c\| \in O(n^2)$. Show that there is a (small) constant c such that

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

(*Hint*: The proof is a slight modification of the standard proof which shows that linear integer programming is NP-complete; see e.g. Hopcroft and Ullman 1979. 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, $\sum_{i=1}^n x_i \leq \sum_{i=1}^n x'_i$. Freeze by an additional equation one variable of x' to be a constant. Repeat the process until the homogeneous system $Dx = 0$ 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 $e(d) \in 2^{O(d)}$, which was used in the proof of Theorem 12.2.1.)

Section 12.4

12.4.1 Consider the reduction in the proof of Lemma 12.3.12. Give an estimation for the length d of the word equation and thereby for an upper bound of $e(\mathscr{B})$. Define another reduction where the denotational length of the resulting word equation becomes smaller. This also improves the estimation for $e(\mathscr{B})$. Give a third estimation for $e(\mathscr{B})$ based on the techniques presented in Section 12.2.
(*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 $y = y'$.)

12.4.2 According to Kościelski and Pacholski 1996, Theorem 4.8 the lower bound for $e(c(S), d)$ given in Example 12.2.3 can be refined. Consider the following equation with $k = 5$.

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

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

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 $1 \leq i \leq n$.)

Notes

A systematic study of equations in free monoids was initiated in the Russian school by A. A. Markov in the late 1950s in connection with Hilbert's Tenth Problem; see Hmelevskiĭ 1971, Makanin 1981. The connection is based on the fact that the set of matrices having nonnegative integer coefficients and determinant 1 form a free monoid inside the special linear group $SL_2(\mathbb{Z})$. 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 $\{a, b\}$ with $\Omega = \{x_1, \ldots, x_n\}$. Replace each variable $x_i \in \Omega$ by a matrix

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

where α_{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 α_{ij}. It is clear that the equation $L = R$ has a solution if and only if the following Diophantine system has a nonnegative solution:

$$\begin{aligned} \alpha_{i1}\alpha_{i4} - \alpha_{i2}\alpha_{i3} &= 1, \quad i = 1, \ldots, n, \\ P_j &= Q_j, \; j = 1, \ldots, 4. \end{aligned}$$

The satisfiability problem of word equations becomes thereby a special instance of Hilbert's Tenth Problem: the satisfiability problem of Diophantine equations. The hope of Markov was to prove the unsolvability of Hilbert's Tenth Problem using this reduction. This hope failed; the unsolvability of Hilbert's Tenth Problem was shown in 1970 by Matiyasevich using an entirely different approach; see Matiyasevich 1993. The solvability of word equations is due to Makanin 1977. It is the subject of the present chapter. However, the reduction from word equations to

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

Diophantine equations is still very useful. For example it yields a simple proof of the Ehrenfeucht conjecture; see Chapter 13 for details.

A consequence of Makanin's result is the decidability of the existential theory of concatenation. The method is given in Subsection 12.1.6. The decidability of the existential theory is close to the borderline to undecidability. By Marchenkov 1982 and by Durnev 1995 it is known that the positive $\forall\exists^3$-theory of concatenation is unsolvable; see also the survey paper of Durnev 1997. Durnev 1974 and Büchi and Senger 1988 defined length predicates such that adding these predicates yields an undecidable existential theory of concatenation. The latter article also shows that equal length is not existentially definable by word equations. The decidability of word equations with an additional equal-length predicate is still an open problem. For more details about the expressibility of languages and relations by word equations see Karhumäki et al. 2000.

A few partial results about the decidability of word equations were known quite early. The fact that a disjunction of two equations can be replaced by a single equation was shown by Büchi in the mid-1960s, but his proof was published only much later; see Büchi and Senger 1986/7. In 1964 and 1967 Hmelevskiĭ found a positive solution for the cases with two and three variables respectively; see Hmelevskiĭ 1971. Other special cases were solved by Plotkin 1972 and Lentin 1972. In the case of two variables Charatonik and Pacholski 1993 analyzed Hmelevskiĭ's work by proving a polynomial-time bound on his algorithm. Their estimation about the degree of the polynomial was rather rough and extremely high. Ilie and Plandowski 2000 lowered the estimation on the degree down to 6 by giving a quadratic bound on the length of the minimal solution. In the case where each variable occurs at most twice, i.e., in the case of quadratic systems, there is a linear-time algorithm for the satisfiability problem, once the lengths for the solutions of variables are fixed and their binary representation is part of the input; see Robson and Diekert 1999. The linear space algorithm for this problem without fixing the lengths appeared in Matiyasevich 1968; the main result of that paper is however a quite different way to reduce word equations with additional conditions on equality of length of some words to Diophantine equations.

After Makanin presented his result in 1977 other questions became central. Makanin 1979 has shown that the rank of an equation is computable; see also Pécuchet 1981. The original article of Makanin is rather technical. Subsequently other presentations with various improvements were given; let us refer to Jaffar 1990, Schulz 1992a, 1993, Gutiérrez 1998b. The present chapter is along this line. A brief survey on equations in words can be found in the paper of Perrin 1989. Further 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 two volumes in the Springer Lecture Notes series dedicated to word equations and related topics: Schulz 1992b and Abdulrab and Pécuchet 1993. Makanin's algorithm was implemented in 1987 at Rouen by Abdulrab; see Abdulrab and Pécuchet 1990.

The inherent complexity of the satisfiability problem of word equations with constants is not yet understood. The lower bound is NP-hardness, simply because linear integer programming (in unary notation) is a special instance and the latter problem is NP-hard. The satisfiability problem of word equations also remains NP-hard for a single quadratic equation. On the other hand, the exponent of periodicity is only linear for quadratic systems; see Diekert and Robson 1999, and it is believed that at least quadratic systems can be solved in NP. In fact, a conjecture of Plandowski and Rytter 1998 claims NP-completeness as the complexity bound for general word equations with constants. The development toward this conjecture over the past few years is somewhat unexpected since a first analysis of Makanin's algorithm done in the works of Jaffar and Schulz showed a 4-NEXPTIME result, only. By Kościelski and Pacholski 1996, Corollary 4.6 this went down to 3-NEXPTIME and then to 2-EXPSPACE during the work on the present chapter. The final version of this chapter uses another improvement due to Gutiérrez 1998a; see Lemma 12.3.14. It shows that the space requirement for Makanin's algorithm does not exceed EXPSPACE. This is the statement of Theorem 12.4.2. It is still the smallest space requirement for a full implementation of Makanin's algorithm.

However, in 1999 Plandowski found a new way for solving word equations whichis independent of Makanin's work and which led to polynomial space. He obtained his result in two consecutive papers which both appeared in 1999: Plandowski 1999a showed that the satisfiability problem for word equations is in NEXPTIME. This is based on a result due to Plandowski and Rytter 1998, which shows that the minimal solution of a word equation is highly compressible in terms of Lempel–Ziv encodings and by a nontrivial combinatorial argument showing that the length of a minimal solution is at most doubly exponential in the denotational length of the equation. The NEXPTIME algorithm is to guess such an encoding of a minimal solution and to verify in deterministic polynomial time that the guess actually corresponds to some solution. Moreover, it is conjectured that the length of a minimal solution is at most exponential in the denotational length of the equation. If this is true, then the Lempel–Ziv encoding will have polynomial length and the satisfiability problem for word equation with constants will become NP-complete. So it might be that the trivial lower bound

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

of NP-hardness already matches the upper bound, which is exactly the conjecture mentioned above. A counterexample to NP-completeness would imply the existence of a family of solvable word equations over a two-letter alphabet where the lengths of minimal solutions grow faster than an exponential function.

Plandowski 1999b showed that the satisfiability problem is in PSPACE. One important ingredient of his work is to use data compression in terms of exponential expressions. It is an interesting open problem whether the use of data compression could also lower the complexity bound in Makanin's method from exponential space down to polynomial space.

This chapter dealt with word equations having rational constraints. In this form the satisfiability problem becomes PSPACE-hard, simply because we may encode the intersection problem for rational languages, and the latter problem is known to be PSPACE-complete by Kozen 1977. Extending Plandowski's method Rytter has stated a PSPACE-completeness result for the satisfiability problem for word equations with rational constraints see Plandowski 1999b, Thm. 1.

Another surprising consequence of Plandowski's work is the dramatic improvement for solving equations over free groups. Let us first recall some background. Word equations in the framework of combinatorial group theory were introduced by Lyndon 1960; see Lyndon and Schupp 1977 for a standard reference. The corresponding notion of quadratic equation plays an important role in the classification of closed surfaces, and basic ideas how to solve quadratic equations go back to Nielsen 1918. The general satisfiability problem for equations with constants in free groups was shown to be decidable by Makanin 1982 and Makanin 1984. Razborov 1984 presented an algorithm which generates all solutions to a given equation. Let us also refer to the survey given by Razborov 1994. Makanin's method for group equations turned out to be even more complicated than in the word case, it is much more involved. Its complexity has been investigated by Kościelski and Pacholski 1998. The authors define the notion of abstract Makanin algorithm and they show that this abstract scheme is not primitive recursive. Therefore it was widely believed that the inherent complexity of the satisfiability problem in the group case is much higher than in the word case. However, there were hints that this was perhaps misleading: Using a result of Merzlyakov 1966 it has been shown by Makanin 1984 that the positive theory of equations in free groups is decidable whereas it was known to be undecidable in the word case. This contrast does not fit well to the assumption that the existential theory over free groups is much harder than over free monoids. And indeed, Gutiérrez 2000 achieved an extension

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press

of Plandowski's method such that it became applicable to the situation in free groups. As in the word case, the existential theory of equations in free groups is in PSPACE. Consequently, a nonprimitive recursive has been replaced by some polynomial space bounded algorithm. Finally, it became possible to cope with rational constraints in free groups. Diekert, Gutiérrez, and Hagenah 2001 have shown that the satisfiability problem for equations with rational constraints in free groups is PSPACE-complete, too.

An ongoing direction of research is to extend Makanin's result beyond free monoids and free groups. We briefly list some of the known results. For example, the main result of Diekert et al. 2001 is in fact a statement about free monoids with involution. This was used when the existential theory of equations in plain groups was shown to be decidable by Diekert and Lohrey 2001, thereby solving an open problem of Narendran and Otto 1997. According to Haring-Smith 1983 a group is called plain, if it is a free product of a finitely generated free group and finitely many finite groups. The class of plain groups is contained in the class of hyperbolic groups, which was introduced by Gromov 1987, and furthermore it is known that the existential theory of equations in torsion-free hyperbolic groups is decidable by Rips and Sela 1995. The intersection of plain groups and of torsion-free word hyperbolic groups is the class of free groups. It is strongly conjectured that the existential theory of equations is decidable in the whole class of hyperbolic groups.

On the other hand, if we move to free inverse semigroups, then the existential theory becomes undecidable; see Rozenblatt 1982, 1985. The situation improves if we wish to include partial commutation. Free partially commutative monoids are also called trace monoids. They are a tool to study some phenomena in concurrency theory; see Mazurkiewicz 1977 and Diekert and Rozenberg 1995 for a general reference. Matiyasevich 1997 has shown that the satisfiability problem of trace equations is decidable; see also Diekert, Matiyasevich, and Muscholl 1999. Diekert and Muscholl 2001 generalized this result to trace monoids with involution and the corresponding result in free partially commutative groups became a corollary. Free partially commutative groups are also calledgraph groups in mathematics; see e.g. Droms 1985, 1987a, 1987b.

The comments above show that the work on word equations led to remarkable results with progress all through the years and many connections to other fields. Makanin's deep insight in the combinatorics on words has been a basis and a source for an active area of research.

https://doi.org/10.1017/CBO9781107326019.013 Published online by Cambridge University Press