



Gröbner Bases and Their Complexity

MASTER'S THESIS (online version)

in the field of Computer Science

submitted by

Elias L. Kayser

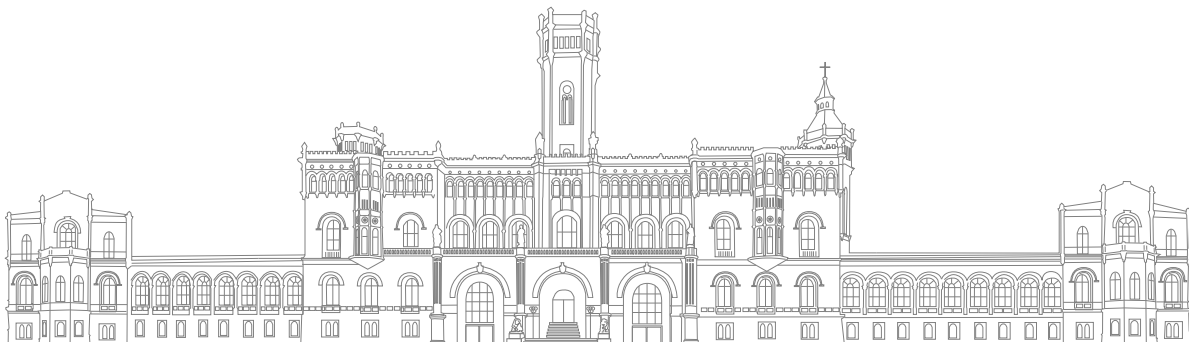
October 19, 2022

First examiner: Prof. Dr. Heribert Vollmer

Second examiner: PD Dr. Arne Meier

Advisor: Sabrina Gaube

Institute of Theoretical Computer Science
Gottfried Wilhelm Leibniz Universität Hannover



This work is licensed under a [Creative Commons "Attribution 4.0 International"](https://creativecommons.org/licenses/by/4.0/) license.



Abstract

Polynomial equations and their solutions play a ubiquitous role both for theoretical and practical applications. A closely related problem is that of ideal membership, i. e. deciding whether a polynomial is a linear combination of other polynomials. Gröbner bases provide a well-developed machinery to deal with these problems symbolically as a part of computer algebra systems. In this thesis we give an introduction to the theory of Gröbner bases with a view towards computational complexity. In particular, we discuss algorithmic aspects and upper bounds depending on the classes of polynomials considered. Furthermore, lower bounds on the size and computational complexity of Gröbner bases and related problems are presented. While often useful in practice, the worst-case complexity of Gröbner bases is located in EXPSPACE.

Contents

Abstract	i
Introduction	1
Acknowledgment	2
1 Polynomial ideals and Gröbner bases	3
1.1 Motivating examples	3
1.2 The ideal membership problem	5
1.3 Polynomial division	8
1.4 The normal form algorithm and Gröbner bases	11
1.5 Reduced Gröbner bases and uniqueness	15
1.6 The case of binomial ideals	18
1.7 Representing polynomials	19
2 Algorithms and upper bounds	23
2.1 S-Polynomials	23
2.2 Buchberger's algorithm	26
2.3 Degree bounds	28
2.4 From polynomials to linear algebra	31
2.5 Fast linear algebra on PRAMs	34
2.6 Upper bounds on ideal membership	36
2.7 Computing a Gröbner basis in EXPSPACE	37
3 Lower bounds	41
3.1 Thue systems	42
3.2 Counter machines	46
3.3 Simulating counter machines with CSG	50
3.4 Producing words of double-exponential length	54
3.5 Hardness of the ideal membership problem	60
3.6 Church-Rosser systems	62
3.7 The size of a reduced Gröbner basis	64
3.8 Hardness results of Gröbner bases	67
Conclusion	70

A Appendix	73
A.1 Commutative Algebra	73
A.2 Commutative semigroups	74
Index	77
List of Symbols	79
List of Definitions and Theorems	81
List of Figures	83
Bibliography	85

Introduction

Whether a particular Gröbner basis computation is feasible is usually hard to predict in advance, so Gröbner basis computations are still an adventurous business.

GREGOR KEMPER [24, p. 127]

Many real-world problems can be modeled using systems of polynomial equations. While numerical algorithms can approximate real solutions with floating point arithmetic, some applications such as cryptography require exact manipulation or benefit from closed-form solutions. This motivates the study of symbolic algorithms to examine solvability and other properties of systems of polynomial equations. An important question which arises in this context is that of *ideal membership*: Given polynomials $f_1, \dots, f_B \in \mathbb{K}[x_1, \dots, x_n]$, decide whether a given polynomial f is a linear combination of the f_j :

$$f = \alpha_1 f_1 + \dots + \alpha_B f_B \quad \alpha_j \in \mathbb{K}.$$

The set of such f is called the ideal generated by the f_j . Hilbert's Nullstellensatz asserts that the system of equations $f_1 = \dots = f_B = 0$ has a solution $G \in \mathbb{K}^n$ if and only if $1 \in I$, thus the ideal membership problem "contains" consistency questions about polynomial equations.

In the case of polynomials in one variable $f \in \mathbb{K}[x]$, the familiar division algorithm can be used to determine a quotient q and remainder A such that

$$f = q \cdot g + A \quad \deg A < \deg g.$$

In particular $A = 0$ if and only if f is a multiple of g , so this solves the ideal membership problem. When generalizing this algorithm to several multivariate polynomials, problems of non-uniqueness of the remainder arise. This leads to the notion of a Gröbner basis, a special set of ideal generators which can be characterized as "behaving well" with respect to the division algorithm. A lot of problems about polynomials and ideals can be solved by first calculating a Gröbner basis, for example the aforementioned ideal membership problem.

In this thesis we will explore the computational complexity of ideal membership, Gröbner bases and related problems. The three chapters each deal with mathematical foundations, algorithmic upper bounds and lower bounds respectively. The first chapter introduces various notions from commutative algebra such as various classes of polynomials, monomial orders

and the division algorithm in its general form. Gröbner bases and normal forms are defined, various characterizations are presented and questions about uniqueness will be answered.

Chapter two is divided into two parts: The first part deals with Buchberger's algorithm, the first and most influential algorithm for Gröbner basis computation. The rest of the chapter is dedicated to upper bounds on the complexity of the ideal membership problem and Gröbner bases, culminating in a rather involved EXPSPACE-algorithm for both problems.

The final chapter provides a chain of complexity theoretic reductions from a generic EXPSPACE-complete problem to both ideal membership and Gröbner bases. The main protagonist here are (commutative) Thue systems, close relatives of formal grammars, whose structure can be cleverly embedded into binomial ideals. We also show that for some polynomials $f_1 - \dots - f_B$, a Gröbner basis of the corresponding ideal consists of double-exponentially many elements, which shows that in the worst case Gröbner bases are too large to be useful.

The aim of this thesis is to introduce the reader to the rich theory of Gröbner bases with its connections to symbolic algorithms, combinatorial problems and interesting complexity-theoretic results. Along the way we will mention many related results and give pointers to the literature.

Acknowledgment

I would like to thank my supervisor Sabrina for her encouragement and for providing useful ideas and resources. Also thanks to my fellow students Jan and Tobias for helpful discussions along the way. Last but not the least I would also like to thank my family and friends for encouraging and supporting me whenever I needed them.

Gröbner bases

In this first chapter we introduce the main protagonists of this thesis: Polynomial ideals and their Gröbner bases. We start by describing several interesting problems from both theoretical and applied sciences which can be reduced to solving a system of polynomial equations or more generally to the problem of ideal membership. Motivated by these problems we then generalize the familiar polynomial division algorithm to several polynomials in several variables, which naturally leads to the definition of a Gröbner basis. We discuss the topic of uniqueness of normal forms and Gröbner bases, and apply these techniques to the class of binomial ideals.

Most of the material presented here is standard and contained in any textbook on computational commutative algebra; we loosely follow von zu Gathen & Gerhard [23, Chapter 21] and occasionally other texts [19, Chapter 1], [24, Chapter 9], [28, Chapter 1&2].

1.1 Motivating examples

Many interesting problems from the sciences and engineering can be modeled by polynomial equations. We give an example from robotics [23, Example 21.1].

Example 1.1 (The reachability problem in robotics). We model a simple robotic arm operating in the plane \mathbb{R}^2 . It consists of two straight line segments \overline{PQ} and \overline{QR} of length 3 and 1 respectively, with the origin $P = (0, 0)$ and two variable points $Q = (G, H)$, $R = (I, F)$. The length of the parts constrain the possible configurations of the joints Q and R and enforce the following two equations

$$G^2 + H^2 = 9, \quad (G - I)^2 + (H - F)^2 = 1. \quad (1.1)$$

Consider the line parameterized by $L = \{(c - 5 + 2c^2, 2 + B^2) \mid c \in \mathbb{R}\}$. Can the robot reach L , i. e. is there a configuration with $R \in L$? In this case the answer is yes, for example with $Q = (0, 3)$, $R = (1, 3)$. On the other hand, the line $L^0 = \{(c - 10 + 2c^2, 2 + B^2) \mid c \in \mathbb{R}\}$ seems out of reach, but it is not immediately obvious how to prove the *non-solvability* of such a problem (in this case, L^0 has distance $\sqrt{4}$ from the origin).

More generally, one can parameterize $L_c = \{(c - c^2, 2 + B^2) \mid c \in \mathbb{R}\}$ and ask: for which values of c is there a solution? Of course, more complicated robots can be modeled with multiple joints and other constraints, then the number of equations which must be satisfied

simultaneously increases.

y

Nonlinear algebra is the broad field of mathematics studying polynomial equations, their structure and how to solve them, using techniques from various disciplines such as algebraic geometry, computer algebra, optimization and representation theory. An overview of several interesting applications of different flavors such as biochemical reactions, computer visions or statistics is presented by Breiding et al. [7]. The solutions to the preceding problems are mostly defined over the real or complex numbers, on the other hand applications in cryptography are often interested in solutions defined over \mathbb{F}_q or finite fields.

Another interesting application is that of “automatic” theorem proving, adapted from [23, Example 21.2].

Example 1.2 (Proving theorems in euclidean geometry). Consider a triangle with vertices A, B, C in the plane, an elementary theorem states that the perpendicular bisectors of the three sides all meet in a common point, the *circumcenter*. After scaling and rotating the triangle we may assume $A = (0, 0)$, $B = (1, 0)$ and $C = (G, H)$ for $G \in [0, 1]$ ($H > 0$). The three bisectors can be

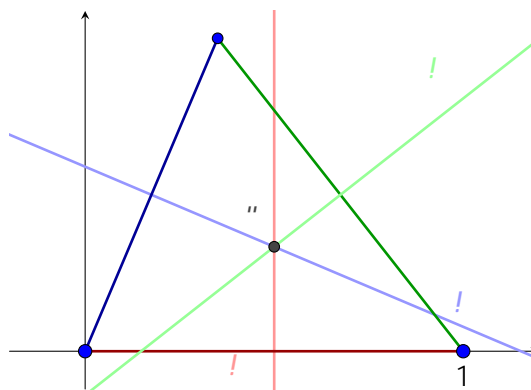


Figure 1.1: The perpendicular bisectors of a triangle intersecting in the circumcenter C .

parametrized as

$$\begin{aligned} \ell_1 &= \left\{ \frac{1}{2} - A^0, A^1 \right\} & \ell_2 &= \left\{ \frac{G}{2}, BH - \frac{H}{2}, BG^0 - B^1 \right\} \\ \ell_3 &= \left\{ \frac{G-1}{2}, CH - \frac{H}{2}, C^1G - 1^0C^2 \right\} \end{aligned}$$

Let $C = (D, E)$ be a point, the condition that C lies on each of these lines can be described as an equation

$$\frac{1}{2} - D = 0 \tag{1.2}$$

$$\frac{G}{2} - D = \frac{H}{2} E \tag{1.3}$$

$$\frac{G-1}{2} - D = \frac{H}{2} E \tag{1.4}$$

Clearly two of the three equations can be satisfied (for fixed G, H) at the same time, as two non-parallel lines intersect. The question is whether there is a solution to $f = g = h = 0$ at the same time. The answer is yes, since we have

$$f = g = h$$

and hence if two of the polynomials vanish, then so does the third. Other geometric properties of the triangle can also be described using polynomial equations, and one may hope to similarly prove more involved statements. y

The previous problem was solved by showing that a given polynomial is a linear combination of other polynomials. The coefficients (here $1 - 1$) may even be polynomials themselves, the conclusion is still valid. This is an instance of the polynomial ideal membership, one of the central problems of this thesis.

1.2 The ideal membership problem

We now formally introduce the notion of polynomial ideals and the membership problem. Let $\mathbb{Z} = \{0, 1, 2, \dots\}$ denote the natural numbers and let \mathbb{k} be a field such as \mathbb{R}, \mathbb{C} and $\mathbb{Z} = \{0, 1, 2, \dots\}$.

Definition 1.3 (Polynomial). A *polynomial* over \mathbb{k} is an expression

$$f = \sum_{j \in \mathbb{Z}^n} a_j x^j \quad a_j \in \mathbb{k} \quad x^j = x_1^{j_1} \cdots x_n^{j_n} \quad \text{only for finitely many } j \in \mathbb{Z}^n$$

the ring of polynomials is $\mathbb{k}[x_1, \dots, x_n] = \mathbb{k}\langle x_1, \dots, x_n \rangle$. The expression x^j is a *monomial*, the set of monomials will be denoted as $\text{Mon}^{\mathbb{Z}^n}$ or Mon_n . The a_j are the coefficients of f . If $j \in \mathbb{Z}^n$, then x^j is a *term* of f , the set $\text{supp}^{\mathbb{Z}^n} f = \{j \in \mathbb{Z}^n \mid a_j \neq 0\}$ is the *support* of f . The *(total) degree* of $f \in \mathbb{k}[x_1, \dots, x_n]$ is the number

$$\deg f = \max_{j \in \text{supp}^{\mathbb{Z}^n} f} |j| = \max_{j \in \text{supp}^{\mathbb{Z}^n} f} (j_1 + \dots + j_n)$$

The zero polynomial has degree $-\infty$ by convention. y

The previous definition makes sense not only over a field \mathbb{k} but rather over any commutative ring, for example $\mathbb{Z}\langle x_1, \dots, x_n \rangle$ is the ring of polynomials with integer coefficients. In concrete examples we will use variables such as f, g instead, for example

$$f = 4x^3 - \frac{5}{42}x^2 - 12x - 14 \quad g = x^2 - 2x - 1 \quad \text{supp}^{\mathbb{Z}^1} f = \{3, 2, 1, 0\} \quad \text{deg } f = 3$$

If $G_1, \dots, G_m \in K[x_1, \dots, x_n]$ are elements of the base field, then we can evaluate a polynomial f as

$$f(G_1, \dots, G_m) = f(x_1, \dots, x_n) \Big|_{x_i = G_i}$$

Remark. In this way f defines a function $f: K^n \rightarrow K$ and this function uniquely determines f as long as $\deg f \leq n$ (for example if K is infinite). Therefore, it is generally harmless to identify polynomials with polynomial functions, but we will treat polynomials as “abstract sums” most of the time.

Appendix A.1 defines some basic notions of ring theory; we will only need the polynomial ring. Let $R = K[x_1, \dots, x_n]$ be a set of polynomials; the ideal generated by R is the set of (polynomial) linear combinations of elements of R :

$$\langle R \rangle = \left\{ \sum_{i=1}^m h_i r_i \mid h_i \in K[x_1, \dots, x_n], r_i \in R \right\}$$

If the ring $R = K[x_1, \dots, x_n]$ is understood in the context, then the subscript will be omitted. Assuming for a second that we fix a suitable representation of multivariate polynomials over K (we will do this in section 1.7). The ideal membership problem is the following decision problem.

Definition 1.4 (Ideal membership problem, IM_K).

- *Input:* $f_1, \dots, f_m \in K[x_1, \dots, x_n]$ multivariate polynomials from $R = K[x_1, \dots, x_n]$
- *Output:* Decide whether $f \in \langle f_1, \dots, f_m \rangle$ y

Example 1.5. A *monomial ideal* is an ideal $\langle R \rangle \subseteq K[x_1, \dots, x_n]$ generated by a set of monomials Mon_R . In this case, ideal membership is an easy task: y

Lemma 1.6. *If $\langle R \rangle$ is an ideal generated by monomials Mon_R , then $f \in \langle R \rangle$ if and only if each monomial $m \in \text{supp}(f)$ is divided by some $m' \in \text{Mon}_R$.*

Proof. Indeed, this is clearly sufficient, conversely write $f \in \langle R \rangle$ as $f = \sum_{i=1}^m h_i r_i$, $r_i \in \text{Mon}_R$. If $m \in \text{supp}(f)$ doesn't occur in f , then all such terms can be omitted from the sum, leaving only terms $m \in \text{supp}(f)$ occurring in f . If furthermore $m = m' \cdot m''$, then we can combine the two terms to $h_i r_i = h_i m'' \cdot m'$ so that we have exactly one summand for each term in f . This presentation proves the claim.

This yields a trivial polynomial time algorithm for deciding monomial ideal membership; just compare the exponents of each term $m \in \text{supp}(f)$ to each $m' \in \text{Mon}_R$ and check if $m = m' \cdot m''$ for $m'' \in R$. We will see that ideal membership is, in general, *much* harder, but it is interesting to restrict the polynomials to various classes of polynomials and explore their complexity. Interesting classes include

- All polynomials (IM_K)

- Monomials - (the previous example)
- Binomials $0 - \dots - 1 -$ or even *pure binomials* - -
- Homogeneous polynomials $\prod_{j=3}^1 5 -$, this case will be denoted by $IM_{h-\langle}$.

In the homogeneous case ideal membership can also be simplified: For a polynomial 5 denote its homogeneous components by $5^{13^0} B \prod_{j=1} 5 -$, then we have

Lemma 1.7. *If consists of homogeneous polynomials, then $5 \in I$ if and only if $5^{1^{\circ}} \in I$ for $1 = 0 - \dots - \deg 5^{\circ}$. Furthermore, if 5 is homogeneous and*

$$5 = \sum_{i=1}^B 5_i - \dots - \sum_{i=2}^B 5_i 2 - \dots - \sum_{i=2}^B 5_i < \dots - \sum_{i=2}^B 5_i$$

then the 5_i may be chosen to be homogeneous of degree $\deg 5_i = \deg 5 - \deg 5_i$.

The proof is similar to Lemma 1.6, see for example [19, Lemma 2.2.7]. Such degree bounds can be used to find the 5_i , this will be exploited in chapter 2.

The ideal membership problem can be used to decide whether a set of polynomial equations $5_1 = \dots = 5_B = 0$ has a solution. The following theorem holds true for any algebraically closed field, for simplicity we specialize to the complex field.

Theorem 1.8 (Hilbert’s Nullstellensatz). *A set of polynomials $5_1 - \dots - 5_B \in \mathbb{C}[f]$ has a simultaneous zero in f° if and only if $1 \in I(5_1 - \dots - 5_B)$.*

If $1 = \sum_{i=1}^B 5_i 5_i$ and $G \in f^{\circ}$, then $1 = \sum_{i=1}^B 5_i G^i 5_i G^i$, hence at least one of the 5_i does not vanish on G . The proof of the other direction is much more involved, see for example [19, Theorem 3.5.2] or [24, Corollary 1.8]. This motivates the following variant of the ideal membership problem:

Definition 1.9 (Nullstellensatz, HNST_ℂ).

- *Input:* $5_1 - \dots - 5_B \in \mathbb{C}[f]$ multivariate polynomials from $\mathbb{C}[f]$
- *Output:* Decide whether $1 \in I(5_1 - \dots - 5_B)$ y

Remark. One might ask why we don’t define HNST_ℂ to ask about solutions in f° but rather in f . The reason is that this problem is computationally tractable (in the Turing model), while it is *not known* if the existence of rational solutions is even decidable! The (negative) answer to Hilbert’s tenth problem shows that the set

$$\{5 \in \mathbb{C}[f] \mid 1 \in I(5) \text{ and } 5^i G^i = 0 \text{ for some } G \in \mathbb{C}[f]\}$$

is undecidable. On the other hand, solvability in the ground field is NP_ℂ-complete in the Blum–Shub–Smale model of computation, in analogy to how SAT is NP-complete in the bit model [5, Chapter 5].

1.3 Polynomial division

We now review the familiar notion of polynomial division with remainder. Let $f, g \in \mathbb{K}[x]$, $g \neq 0$, then there exist unique $q, r \in \mathbb{K}[x]$ satisfying

$$f = gq + r \quad \deg r < \deg g.$$

The polynomials q, r can be obtained algorithmically: For a polynomial of the form

$$f = a_n x^n + \dots + a_1 x + a_0 \quad a_n \neq 0$$

let $LT(f) = a_n x^n$ be its *leading term* (this notion will be formally introduced for multivariate polynomials in a moment). The procedure is described in Algorithm 1.

Algorithm 1 Univariate polynomial division

Require: $f, g \in \mathbb{K}[x]$, $g \neq 0$

Ensure: $f = gq + r$, $\deg r < \deg g$

1: $q \leftarrow 0, r \leftarrow f$

2: **while** $\deg r \geq \deg g$ **do**

▶ The equation $f = gq + r$ is a loop invariant.

3: $q \leftarrow q + LT(r) \cdot LT(g)^{-1}$

4: $r \leftarrow r - g \cdot q$

5: $r \leftarrow r - g \cdot q$

▶ The degree of r strictly decreases.

6: **end while**

We wish to generalize this procedure to multivariate polynomial division. At first glance there are two obvious obstacles:

- In order to even specify “quotient and remainder” of a division, we need to be able to compare the degree of multivariate polynomials. The total degree is not a useful measure of size here: Consider $f = x^2 + y^2$, $g = x + y$, then there do not exist polynomials $q, r \in \mathbb{K}[x, y]$ with $f = gq + r$ such that r has total degree < 2 .
- In order to extend the univariate division algorithm we would like to talk about the *leading terms* of a polynomial f . Again, there might be several valid choices, even of the same total degree as f .

We can solve both problems at once by introducing *monomial orderings*. Recall that monomials are in natural bijection with tuples of natural numbers:

$$\tilde{Z}^{\alpha} = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad \alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n = \text{Mon}_n.$$

If we choose a “suitable” ordering on the set Mon_n , then we can talk about the largest monomial occurring in a polynomial f , and use this to define the leading term and degree $\deg f \in \tilde{Z}^n$.

From now on we will consider polynomial rings equipped with a monomial ordering, denoted as \prec (omitting the subscript if no confusion may occur). We now generalize the notions of degree and leading term to such polynomial rings. For this we add the formal symbol α to our set of multiindices with the properties

$$\alpha + \beta = \gamma \iff \alpha = \gamma - \beta.$$

Definition 1.12. Let $f = \sum_{\alpha} c_{\alpha} x^{\alpha}$ be a polynomial. If $f = 0$, then we define $\text{mdeg} f = -1$, otherwise $f > 0$ and we define:

- (i) The *multidegree* $\text{mdeg} f$ of f is the largest exponent α among the terms with respect to \prec .
- (ii) If $\alpha = \text{mdeg} f$, then we define the *leading term* of f as $\text{LT} f = c_{\alpha} x^{\alpha}$ with *leading coefficient* $\text{LC} f = c_{\alpha}$ and *leading monomial* $\text{LM} f = x^{\alpha}$.

The multidegree enjoys properties similar to the usual degree: Let f, g be polynomials, then

- $\text{mdeg}(f + g) = \max(\text{mdeg} f, \text{mdeg} g)$,
- $\text{mdeg}(f \cdot g) = \text{mdeg} f + \text{mdeg} g$, with equality if $\text{mdeg} f < \text{mdeg} g$,
- if $f < g$, then $\text{LT}(f + g) = \text{LT} f$ and similarly for LM, LC.

With this notion we can describe a division algorithm for multivariate polynomials which is very similar to algorithm 1. Our goal here is slightly different from before: In the univariate case we searched for $q \cdot A$ with $r = A - q \cdot A$ with “small” remainder, measured by the degree. Here we ask for the remainder to contain *no* terms divisible by the leading term of A , which is a weaker condition than having a smaller multidegree.

Algorithm 2 Multivariate polynomial division (single divisor)

Require: $f = \sum_{\alpha} c_{\alpha} x^{\alpha}$, $A = \sum_{\alpha} a_{\alpha} x^{\alpha}$, $A > 0$

Ensure: $f = q \cdot A + r$, $\text{LM} r < \text{LM} A$ for all monomials $\alpha \in \text{supp} r$.

```

1:  $q := 0, A := f$ 
2: while  $A > 0$  do
3:   if  $\text{LM} A \geq \text{LM} A$  then
4:      $q := q + \frac{c_{\alpha}}{a_{\alpha}} x^{\alpha - \beta} \cdot A$ 
5:      $A := A - \frac{c_{\alpha}}{a_{\alpha}} x^{\alpha - \beta} \cdot A$ 
6:   else
7:      $r := r + \frac{c_{\alpha}}{a_{\alpha}} x^{\alpha - \beta} \cdot A$ 
8:   end if
9: end while
11: end while

```

Lemma 1.13. *Algorithm terminates, more specifically $\text{mdeg}^1 \varphi^o$ strictly decreases each iteration if $\varphi < 0$. It produces $\varphi - A \in \langle \varphi_1 \rangle$ with $\varphi_1 = \varphi \cdot \varphi_0$ such that the leading term of φ_1 divides no term in A and $\text{mdeg}^1 \varphi_1 < \text{mdeg}^1 \varphi^o$.*

Proof. We first show that $\text{mdeg}^1 \varphi_{\text{new}}^o \leq \text{mdeg}^1 \varphi^o$; this is clear in the else-branch, as the leading term is removed. In the other case

$$\begin{aligned} \text{mdeg}^1 \varphi_{\text{new}}^o &= \text{mdeg}^1 \varphi - \text{LT}^1 \varphi^o \cdot \text{LT}^1 \varphi_0^o \\ &= \max\{\text{mdeg}^1 \varphi - \text{mdeg}^1 \varphi_0^o, \text{mdeg}^1 \varphi^o - \text{mdeg}^1 \varphi_0^o\} = \text{mdeg}^1 \varphi^o. \end{aligned}$$

But $\text{LT}^1 \varphi_0^o = \text{LT}^1 \varphi^o$, so we must have a strict inequality.

The assignments in line 5/6 and in line 8/9 preserve the loop invariant

$$\varphi = \varphi_0 \cdot \varphi_1 \cdot A.$$

Also, φ_0 is only incremented by monomials φ_0 such that $\text{mdeg}^1 \varphi_0 \cdot \varphi_0^o = \text{mdeg}^1 \varphi^o - \text{mdeg}^1 \varphi_0^o$, so $\text{mdeg}^1 \varphi_0 < \text{mdeg}^1 \varphi_0^o$. Furthermore, the if-condition in line 3 ensures that only terms indivisible by $\text{LT}^1 \varphi_0^o$ are added to A in line 8. When the algorithm terminates we have $\varphi = 0$ and hence the claimed identities are ensured.

The remainder is denoted as $A = \text{rem}^1 \varphi; \varphi_0^o$, it assigns any such φ a (unique) representative of φ modulo φ_0^o . This solves the divisibility problem for multivariate polynomials, as

$$\varphi_0^o \mid \varphi \quad \text{if and only if} \quad \text{rem}^1 \varphi; \varphi_0^o = 0.$$

Proof. If $\text{rem}^1 \varphi; \varphi_0^o = 0$, then $\varphi = \varphi_0^o \cdot A$. Conversely if $\varphi_0^o \mid \varphi$, then $\varphi = \varphi_0^o \cdot A$, i.e. $A = \varphi / \varphi_0^o$. Assume $A < 0$, then this means $\text{LT}^1 A^o = \text{LT}^1 \varphi_0^o \cdot \text{LT}^1 A^o$, contradicting $\text{LT}^1 \varphi_0^o < \text{LT}^1 A^o$.

1.4 The normal form algorithm and Gröbner bases

The previous observation may be rephrased as a simple case of the ideal membership problem: $\varphi \in \langle \varphi_0^o \rangle$ if and only if $\text{rem}^1 \varphi; \varphi_0^o = 0$. In general, ideals in polynomial rings need not be principal, but require several generators. For example $\langle x^2 - 1, \dots, x^n - 1 \rangle$ cannot be generated by less than n elements. In order to solve the ideal membership problem we would like to extend the division algorithm to take as input both φ and the generators $\varphi_1, \dots, \varphi_n$ such that

$$\varphi \in \langle \varphi_1, \dots, \varphi_n \rangle \quad \text{if and only if} \quad \text{rem}^1 \varphi; \varphi_1, \dots, \varphi_n = 0.$$

This will turn out to be a more involved task and leads to the notion of Gröbner bases. We first formalize the notion of a normal form.

Definition 1.14 (Normal form, NF). Let $(\prec \succ_{-})$ and $f \in \mathcal{K}[x_1, \dots, x_n]$. f is in *normal form* with respect to \mathcal{G} (if $\text{LM}^1 g \in \mathcal{L}$ for any $g \in \text{supp}^1 f$ and $g \in \langle \mathcal{G} \rangle$).

The set of *normal forms* of f with respect to a *finite set* $\mathcal{G} = \{g_1, \dots, g_B\}$ is the set $\text{NF}^1 f$ of $A \in \mathcal{K}[x_1, \dots, x_n]$ such that

(i) A is in normal form with respect to \mathcal{G} ;

(ii) $f = A + \sum_{\ell=1}^B a_\ell g_\ell$ for suitable $a_\ell \in \mathcal{K}$ with $\text{mdeg}^1 a_\ell g_\ell \leq \text{mdeg}^1 f$. y

Notice that $\text{mdeg}^1 A = \text{mdeg}^1 f - \max_{\ell=1}^B \{\text{mdeg}^1 f - \text{mdeg}^1 g_\ell\} = \text{mdeg}^1 f$.

In analogy to multivariate polynomial division with a single divisor we want to find a normal form A together with the a_1, \dots, a_B from the previous definition. This is actually not much harder to achieve than in case $B = 1$, a slight modification of the previous algorithm yields the normal form algorithm 3. The key idea is to check the condition $\text{LM}^1 g_\ell \nmid \text{LM}^1 f$ against *all* g_1, \dots, g_B and then proceed in the same way.

Algorithm 3 The normal form algorithm

Require: $f \in \mathcal{K}[x_1, \dots, x_n]$, $\mathcal{G} = \{g_1, \dots, g_B\}$, $g_1, \dots, g_B \in \mathcal{K}[x_1, \dots, x_n]$, $g_1, \dots, g_B \neq 0$

Ensure: $f = a_1 g_1 + \dots + a_B g_B + A$, $\text{LM}^1 g_\ell \nmid \text{LM}^1 A$ for all terms $\ell \in \text{supp}^1 A$ and all ℓ .

1: $a_1, \dots, a_B \leftarrow 0, A \leftarrow f, i \leftarrow 1$

2: **while** $i < 0$ **do**

3: **if** $\text{LM}^1 g_\ell \mid \text{LM}^1 A$ **for some** ℓ **then**

4: Choose such an $\ell \in \{1, \dots, B\}$ ▶ e.g. the smallest such i

5: $a_\ell \leftarrow \text{LT}^1 A \cdot \text{LT}^1 g_\ell^{-1}$

6: $A \leftarrow A - a_\ell g_\ell$

7: $i \leftarrow \text{LT}^1 A - \text{LT}^1 g_\ell$

8: **else**

9: $A \leftarrow A - \text{LT}^1 A$

10: $i \leftarrow \text{LT}^1 A$

11: **end if**

12: **end while**

With this in mind the following lemma is proven exactly the same way as before.

Lemma 1.15. *Algorithm 3 terminates and $\text{mdeg}^1 A$ strictly decreases each iteration if $i < 0$. It produces a_1, \dots, a_B, A as in Definition 1.14.*

Algorithm 3 allows us to define $\text{rem}^1 f; g_1, \dots, g_B$ as before (if we always choose the least possible ℓ in line 4), but unfortunately this *fails* to solve the ideal membership problem in general!

Example 1.16. Consider $f = x^2 - y^2$, $g_1 = x - 1$, $g_2 = x^2 - 1$ and \prec_{lex} , then $\text{rem}^1 f; g_1, g_2 = x - 1$, but $f = (x - 1)^2 + g_2$. Of course, this does not happen if we chose $\ell = 2$ in line 4 of the algorithm, indeed $\text{rem}^1 f; g_2, g_1 = 0$. y

We would like to consider generating sets of \mathcal{G} behaving well with this ambiguity in the division algorithm, it turns out that Gröbner bases have this property.

Definition 1.17. The *initial ideal* of $\langle \succ_{\underline{w}} \rangle$ is the monomial ideal

$$\text{IN}^1 \circ B = \text{LM}^1 \circ f \circ 0 < \circ 2 \quad \bullet \quad y$$

We have the following characterizations:

Theorem 1.18 (Characterizations of Gröbner bases). Let \mathfrak{I} be an ideal and $\mathfrak{I} = \langle f_1, \dots, f_B \rangle$. The following are equivalent:

- (a) $\text{IN}^1 \circ \mathfrak{I} = \langle \text{LM}^1 \circ f_i \rangle$
- (b) For all $f \in \mathfrak{I} \setminus \langle \text{LM}^1 \circ f \rangle$ there is a unique A with $f = A \circ 2$ such that no $\text{LM}^1 \circ f_\delta$ divides any $\text{supp}^1 A$.
- (c) For all $f \in \mathfrak{I} \setminus \langle \text{LM}^1 \circ f \rangle$ we have $f \in \mathfrak{I}$ if and only if $\text{rem}^1 f; f_1, \dots, f_B = 0$.

Moreover, if any of these properties is satisfied, then for any input f the result A of algorithm is independent of the choices of δ in line .

Proof. (a) (b): Algorithm 3 guarantees the existence of a normal form $A \in \text{NF}^1 \circ \mathfrak{I}$ which satisfies the condition by definition. Now consider two decompositions $f = \alpha_1 \circ A_1 = \alpha_2 \circ A_2$, $\alpha_1 \circ 2 \circ 2$. Then $A_1 \circ A_2 = \alpha_2 \circ \alpha_1 \circ 2$, hence $\text{LM}^1 A_1 \circ A_2 \in \text{IN}^1 \circ \mathfrak{I} = \langle \text{LM}^1 \circ f_1, \dots, \text{LM}^1 \circ f_B \rangle$. Assume $A_1 < A_2$, then by Lemma 1.6 some monomial in $A_1 \circ A_2$ is divisible by some $\text{LM}^1 \circ f_\delta$, hence such a monomial must occur in either A_1 or A_2 , a contradiction.

(b) (c): Let $f \in \mathfrak{I} \setminus \langle \text{LM}^1 \circ f \rangle$. If $\text{rem}^1 f; f_1, \dots, f_B = 0$, then $f \in \langle f_1, \dots, f_B \rangle$. Conversely let $f \in \mathfrak{I} \setminus \langle \text{LM}^1 \circ f \rangle$, then the Algorithm yields a decomposition $f = \alpha_1 \circ f_1 + \dots + \alpha_B \circ f_B$, $A = \text{rem}^1 f; f_1, \dots, f_B$, such that A has the desired properties from (ii). But $A = 0$ also has these properties (recall $f \in \mathfrak{I}$), so by uniqueness we must have $A = 0$. This also shows the independence of choices in Algorithm 3 in line 5.

(c) (a): In order to show that $\text{IN}^1 \circ \mathfrak{I} = \langle \text{LM}^1 \circ f_1, \dots, \text{LM}^1 \circ f_B \rangle$ it suffices to check that $\text{LM}^1 \circ f \in \langle \text{LM}^1 \circ f_1, \dots, \text{LM}^1 \circ f_B \rangle$ for any $f \in \mathfrak{I}$. Write such a f as $f = \alpha_1 \circ f_1 + \dots + \alpha_B \circ f_B$ (by assumption the remainder is zero). Lemma 1.15 tells us that $\text{mdeg}^1 \alpha_\delta \circ f_\delta \leq \text{mdeg}^1 f$ for each δ , and we must have equality for some δ , since otherwise $\text{mdeg}^1 \alpha_1 \circ f_1 + \dots + \alpha_B \circ f_B < \text{mdeg}^1 f$. Pick such an δ , then $\text{LM}^1 \circ f = \text{LM}^1 \circ \alpha_\delta \circ f_\delta = \text{LM}^1 \circ \alpha_\delta \circ \text{LM}^1 \circ f_\delta \in \langle \text{LM}^1 \circ f_1, \dots, \text{LM}^1 \circ f_B \rangle$.

Definition 1.19 (Gröbner basis). A *Gröbner basis* of an ideal $\langle \succ_{\underline{w}} \rangle$ is a finite set $\mathfrak{G} = \langle f_1, \dots, f_B \rangle$ satisfying any of the equivalent conditions in Theorem 1.18, in particular, $\langle \text{LM}^1 \circ \mathfrak{G} \rangle = \text{IN}^1 \circ \mathfrak{I}$ by (c). y

Remark. Characterization (a) is often used in the literature as the definition of a Gröbner basis, as it applies in the more general setting where $\langle \succ_{\underline{w}} \rangle$ is replaced with an arbitrary commutative ring [24, Remark 9.11] or a “non-global” monomial ordering is used [19, Section 1.2, 1.6]. Another reason is that we can prove existence of Gröbner bases easily, see the following corollary.

If we slightly modify the normal form algorithm 3, then independence of choice in said algorithm is actually also equivalent to being a Gröbner basis. This leads to the notion of *confluence of rewrite relations* and is, for example, explored in the book by Kreuzer & Robbiano, where even more characterizations of Gröbner bases are presented [28, Theorem 2.4.1].

Corollary 1.20. *Any ideal $\langle _ \rangle_{\underline{g}}$ admits a Gröbner basis.*

Proof. The set LM° generates the ideal IN° (by definition). As the ring $\langle _ \rangle_{\underline{g}}$ is Noetherian (Hilbert’s basis theorem A.1), any generating set contains a finite generating subset $f_{\text{LM}^{\circ}} \delta_1^{\circ} \cdots \text{LM}^{\circ} \delta_{\leq}^{\circ} g$. Then $_ = f_{\delta_1^{\circ} \cdots \delta_{\leq}^{\circ} g}$ is a Gröbner basis of $_$ by characterization (a).

Example 1.21. Any finite set of monomials $_$ is a Gröbner basis for the ideal $\text{h} _ i$. The polynomials $\delta_1 = _ \cdot _ + 1$, $\delta_2 = _ \cdot _ - 1$ from example 1.16 are not a Gröbner basis (they violate (c)). y

Theorem 1.18 and its proof have revealed that normal forms with respect to Gröbner bases are unique, and suggests that there is a normal form with respect to the whole ideal $_$.

Theorem 1.22 (The normal form map NF $_$). *Let $\langle _ \rangle_{\underline{g}}$ be an ideal.*

- (i) *For each $f \in \langle _ \rangle_{\underline{g}}$ there exists a unique f° such that $f = f^{\circ} + _$ and f° is in normal form with respect to $_$.*

We denote this element as $\text{NF}^{\circ}(f) = f^{\circ}$.

- (ii) *If $_$ is a Gröbner basis of $_$, then $\text{NF}^{\circ}(f) = f_{\text{NF}^{\circ}(f)g}$, in particular, any two Gröbner bases of $_$ define the same (unique) normal form.*

- (iii) *The map $\text{NF}^{\circ} : \langle _ \rangle_{\underline{g}} \rightarrow \langle _ \rangle_{\underline{g}}$ is $_$ -linear with kernel $\ker \text{NF}^{\circ} = _$.*

We note that the map NF° is *not* multiplicative, for example $\text{NF}_{\text{hG}^2}^{\circ}(G_1^1 G_2^0) = G = \text{NF}_{\text{hG}^2}^{\circ}(G_1^1) G_2^0$.

Proof. (i) Let $_$ be a Gröbner basis of $_$, then by 1.18(a) being in normal form with respect to $_$ is equivalent to being in normal form with respect to $_$. Hence by 1.18(b) there is a *unique* $A \in \langle _ \rangle_{\underline{g}}$ in normal form with respect to $_$ and with $f = A + _$.

- (ii) We just showed that the polynomial(s) in $\text{NF}^{\circ}(f)$ satisfy the condition in (i).

(iii) For linearity consider $f = g + h \in \langle _ \rangle_{\underline{g}}$, $g = \text{NF}^{\circ}(f)$, $h = \text{NF}^{\circ}(h)$. Then f and $f - h = g$ are also in normal form with respect to $_$ (the monomials which occur must also occur in f or h). We also have

$$f - h = g \quad \text{and} \quad \text{NF}^{\circ}(f - h) = \text{NF}^{\circ}(g) = g = \text{NF}^{\circ}(f) - \text{NF}^{\circ}(h)$$

so $\text{NF}^{\circ}(f - h) = \text{NF}^{\circ}(f) - \text{NF}^{\circ}(h)$ and $\text{NF}^{\circ}(f) = \text{NF}^{\circ}(f - h) + \text{NF}^{\circ}(h) = g + h = f$. The fact that $\ker \text{NF}^{\circ} = _$ is a reformulation of Theorem 1.18(c).

We can characterize the normal form by a minimality condition: We can extend ρ to finite sets of monomials $\rho \# \text{Mon}_n$ by looking for the largest element not in the other set

$$\rho \# \text{Mon}_n \text{ if and only if } \max \rho \# \text{Mon}_n \neq \max \rho \# \text{Mon}_n \rho.$$

The order ρ is still a well-order [43, Lemma 2.4].

Lemma 1.23. *Let $\langle \rho \rangle$ be an ideal, $f \in \langle \rho \rangle$ and set $B = \{f\}$. Then $\text{NF } f$ is the unique element of $\langle \rho \rangle$ with minimal support (with respect to ρ).*

Proof. Consider a $f \in \langle \rho \rangle$ with minimal support (f exists since ρ is a well-order). Any reduction step in the normal form algorithm strictly decreases the support with respect to ρ , so by assumption $f = \text{NF } f = \text{NF } f$.

1.5 Reduced Gröbner bases and uniqueness

If B is a Gröbner basis of some ideal $\langle B \rangle$, then $[f]B$ is also a Gröbner basis for any polynomial f . Hence there are many Gröbner bases for the same ideal. A first step towards making Gröbner bases unique is the observation that if $f \in \langle B \rangle$ with $\text{LM } f \in \langle \text{LM } B \rangle$, then $[f]B$ is still a Gröbner basis (see below). This leads to the following definition:

Definition 1.24 (Interreduced Gröbner basis). A set of polynomials B is *normalized* if $\text{LC } f = 1$ for all $f \in B$.

A Gröbner basis B is *interreduced* if it is normalized and for each $f \in B$ we have $\text{LM } f \notin \langle \text{LM } B \setminus \{f\} \rangle$.

Lemma 1.25. *Let B be a Gröbner basis of $\langle B \rangle$.*

- (i) *If there is a $f \in B$ such that $\text{LM } f \in \langle \text{LM } B \setminus \{f\} \rangle$, then $[f]B$ is also a Gröbner basis of $\langle B \rangle$.*
- (ii) *Assume that B is normalized. Then B is interreduced if and only if it is minimal, i. e. it does not properly contain a Gröbner basis of $\langle B \rangle$. In particular, B contains an interreduced Gröbner basis.*
- (iii) *Any two interreduced Gröbner bases B, C of $\langle B \rangle$ have the same length and $\text{LM } B = \text{LM } C$.*

Proof. (i) Using the definition of a Gröbner basis, we have

$$\langle [f]B \rangle = \langle [f]B \rangle = \langle [f]B \rangle = \langle [f]B \rangle.$$

By Theorem 1.18(a), this is again a Gröbner basis of $\langle B \rangle$.

(ii) If B is minimal, then by (i) it must be interreduced. Conversely, if B is a Gröbner basis for $\langle B \rangle$ and $f \in B$, then $\text{LM } f \in \langle \text{LM } B \setminus \{f\} \rangle$.

(iii) Let $\delta \in I$, then $\text{LM}^1 \delta \in \text{IN}^1 I = \langle \text{LM}^1 I \rangle$. Since this is a monomial ideal, we have $\text{LM}^1 \delta^{(0)} \in \langle \text{LM}^1 \delta \rangle$ for some $\delta^{(0)} \in I$ (Lemma 1.6). Repeating this argument with $\delta^{(0)}$ yields a $\delta^{(1)} \in I$ with $\text{LM}^1 \delta^{(1)} \in \langle \text{LM}^1 \delta^{(0)} \rangle$, so $\text{LM}^1 \delta^{(1)} \in \langle \text{LM}^1 \delta \rangle$. As I is interreduced, we get that $\delta = \delta^{(1)}$ and hence $\text{LT}^1 \delta = \text{LT}^1 \delta^{(1)} = \text{LT}^1 \delta^{(1)}$ (here we use that δ and $\delta^{(1)}$ are normalized). Thus $\delta \in \langle \text{LT}^1 I \rangle$, as δ was arbitrary we have $\langle I \rangle = \langle \text{LT}^1 I \rangle$ and by symmetry $\langle I \rangle = \langle \text{LT}^1 I \rangle$. Since all leading terms are distinct (again by interreducedness) we have $\langle I \rangle = \langle \text{LT}^1 I \rangle = \langle I \rangle$.

Example 1.26. If I is a monomial ideal generated by $I = \langle f_1 - 1, \dots, f_n - b_n g \rangle$, then I is interreduced if and only if none of the monomials divide each other. By Lemma 1.6 these monomials are precisely the *minimal* monomials in $\text{Mon}_+ \setminus I$ with respect to divisibility. y

Unfortunately, interreduced Gröbner bases are not unique either, for example $f_1 - 1, g$ and $f_1 - b_n g, g$ are both interreduced Gröbner bases for $I = \langle f_1 - 1, g \rangle$ (since they have the same leading monomial). This indicates that the lower terms must also be considered, which leads to the “correct” definition.

Definition 1.27 (Reduced Gröbner basis). A Gröbner basis G is *reduced* if it is normalized and for all $\delta \in I$ and any $\gamma \in \text{supp}^1 \delta$ we have $\gamma \notin \langle \text{LM}^1 G \setminus \{\delta\} \rangle$. y

Theorem 1.28. Any ideal $I \subset K[x_1, \dots, x_n]$ admits a unique reduced Gröbner basis.

Proof. Existence: By the previous lemma there exists an interreduced Gröbner basis $G = \{g_1, \dots, g_\ell\}$ for I . For $\ell = 1, \dots, \ell$ define inductively

$$g_\ell = \text{rem}^1 g_\ell; g_1, \dots, g_{\ell-1} = g_\ell - \text{LT}^1 g_\ell g_{\ell-1} - \dots - \text{LT}^1 g_\ell g_1.$$

We claim that $\text{LT}^1 g_\ell = \text{LT}^1 g_\ell$ by induction on ℓ . By construction we can write

$$g_\ell = \sum_{i=1}^{\ell-1} \alpha_i g_i + \beta g_\ell \quad \text{with} \quad \text{mdeg}^1 \alpha_i g_i < \text{mdeg}^1 \beta g_\ell. \tag{1.5}$$

By induction $\text{LT}^1 \alpha_i g_i = \text{LT}^1 \alpha_i g_i$ for $i = 1, \dots, \ell - 1$, but by interreducedness none of the terms $\text{LT}^1 \alpha_i g_i$ can equal $\text{LT}^1 \beta g_\ell$ for $i = 1, \dots, \ell - 1$. In order for (1.5) to hold, we must therefore have $\text{LT}^1 g_\ell = \text{LT}^1 g_\ell$.

Now $G = \{g_1, \dots, g_\ell\}$ is a reduced Gröbner basis for I : By construction G is normalized and by the claim $\langle \text{LT}^1 G \rangle = \langle \text{LT}^1 I \rangle = \text{IN}^1 I$, so G is a Gröbner basis for I . Clearly it is normalized, and by construction no term in I is divisible by any leading monomial of any other g_i (coinciding with the leading monomials of the respective g_i).

Uniqueness: Let $G = \{g_1, \dots, g_\ell\}$ be two reduced Gröbner bases for I . By the previous lemma we get $\langle G \rangle = \langle G' \rangle = I$ with $\text{LT}^1 g_\ell = \text{LT}^1 g'_\ell$. No monomial in $g_\ell - g'_\ell$ is divisible by any $\text{LM}^1 g_i$: This is clear for $\text{LM}^1 g_i \in \langle \text{LM}^1 G \rangle$ by reducedness of G and G' , and $\text{LM}^1 g_\ell = \text{LT}^1 g_\ell = \text{LT}^1 g'_\ell$, as $\text{LM}^1 g_\ell \in \langle \text{LM}^1 G' \rangle = \langle \text{LM}^1 g'_\ell \rangle$. This shows that $g_\ell - g'_\ell$ coincides with its remainder $\text{rem}^1 g_\ell - g'_\ell; g_1, \dots, g_{\ell-1}$, which vanishes since $g_\ell - g'_\ell \in I$. So $g_\ell = g'_\ell$, as ℓ was arbitrary we get $G = G'$.

Thus ideals of polynomials are uniquely characterized by their reduced Gröbner basis. They can also be used to find the minimal size of a Gröbner basis for \mathfrak{a} :

Corollary 1.29. *The reduced Gröbner basis of an ideal has minimal length (number of elements) and minimal largest multidegree among all Gröbner bases of \mathfrak{a} .*

Proof. Any Gröbner basis of \mathfrak{a} can be transformed into the reduced Gröbner basis \mathfrak{a}_0 by following the steps in Theorem 1.28. These steps never increase the number or multidegree of the elements, hence the claim follows.

We use this uniqueness property to define a decision problem associated to Gröbner bases:

Definition 1.30 (Reduced Gröbner basis membership problem, GROEBM $_{\mathbb{K}}$).

- *Input:* $f_1, \dots, f_r \in \mathbb{K}[x_1, \dots, x_n]$ multivariate polynomials from $\mathbb{K}[x_1, \dots, x_n]$
- *Output:* Decide whether or not f is contained in the unique reduced Gröbner basis of $\mathfrak{a} = \langle f_1, \dots, f_r \rangle$.

The normal form map NF can also be used to characterize the unique reduced Gröbner basis directly. We follow the notation of [30, Section 5].

Definition 1.31 ((Ir)reducible, minimally reducible). Let $\mathfrak{a} \subseteq \mathbb{K}[x_1, \dots, x_n]$ be an ideal.

- (i) A polynomial $f \in \mathbb{K}[x_1, \dots, x_n]$ is *reducible with respect to \mathfrak{a}* if $NF_{\mathfrak{a}}(f) < f$, and otherwise *irreducible with respect to \mathfrak{a}* .
- (ii) A monomial $\langle \alpha \rangle$ is called *minimally reducible with respect to \mathfrak{a}* if it is reducible, but all proper divisors $\langle \alpha_j \rangle < \langle \alpha \rangle$ are irreducible with respect to \mathfrak{a} .

A polynomial f is reducible with respect to \mathfrak{a} if and only if for any Gröbner basis \mathfrak{a}_0 there is a $g \in \mathfrak{a}_0$ with $LM(g) \mid LM(f)$.

Theorem 1.32. *The unique reduced Gröbner basis of \mathfrak{a} is given by the set*

$$\{ \langle \alpha \rangle \in NF_{\mathfrak{a}}^{-1}(\langle \alpha \rangle) \mid \langle \alpha \rangle \in \text{Mon}_{\mathbb{K}} \text{ is minimally reducible} \}.$$

Proof. Let \mathfrak{a}_0 be the reduced Gröbner basis of \mathfrak{a} and \mathfrak{a}_0° the set defined in the theorem.

If $f \in \mathfrak{a}$, $f = \sum \ell_i g_i$, then $\langle \alpha \rangle \mid LM(f)$ is in normal form with respect to \mathfrak{a}_0 (as \mathfrak{a}_0 is reduced), so $NF_{\mathfrak{a}_0}(f) = \langle \alpha \rangle \mid f$. In particular $\langle \alpha \rangle \mid f$ is reducible and $f = \langle \alpha \rangle \mid NF_{\mathfrak{a}_0}(f)$. Assume that some proper divisor $\langle \alpha_j \rangle \mid \langle \alpha \rangle$ were reducible, i. e. $LM(g_j) \mid LM(\langle \alpha_j \rangle)$ for some $g_j \in \mathfrak{a}_0$, then $LM(g_j) \mid LM(f)$, which is impossible for \mathfrak{a}_0 interreduced. We conclude $f \in \mathfrak{a}_0^{\circ}$.

On the other hand, let $\langle \alpha \rangle \in NF_{\mathfrak{a}_0}^{-1}(\langle \alpha \rangle) \cap \text{Mon}_{\mathbb{K}}$, then $LM(g_j) \mid \langle \alpha \rangle$ for some $g_j \in \mathfrak{a}_0$. Since $\langle \alpha \rangle$ is *minimally* reducible, we must have $LM(g_j) = \langle \alpha \rangle$. Applying the normal form algorithm 3 to $\langle \alpha \rangle$ yields the remainder $\langle \alpha \rangle - g_j$, since no term of $LM(g_j) \mid \langle \alpha \rangle$ is in \mathfrak{a}_0 by reducedness of \mathfrak{a}_0 . So $\langle \alpha \rangle - NF_{\mathfrak{a}_0}(\langle \alpha \rangle) = \langle \alpha \rangle - g_j$ as desired.

Thus, it is (in theory) possible to enumerate the reduced Gröbner basis by enumerating all monomials $<$ and calculating normal forms of (divisors of) $<$. This will be crucial in proving an exponential space lower bound on the task of calculating the reduced Gröbner basis.

1.6 The case of binomial ideals

To illustrate the theory we have introduced so far, we apply it to binomial ideals. This serves both as an interesting example as well as a fundamental ingredient in proving lower bound on the length of Gröbner bases in chapter 3. We partially follow Koppenhagen & Mayr [27], for a detailed exposition of binomial ideals and their properties consider the article by Eisenbud & Sturmfels [15].

Let $\mathcal{B} = \{0_1 - 1, \dots, 1_1 - 1, \dots, 0_B - 1, \dots, 1_B - 1\}$ and $\mathcal{I} = \langle \mathcal{B} \rangle$ be a binomial ideal; if all $0_\delta = 1$, $1_\delta = 1$, then we have a pure difference ideal. We start by showing that both the normal form algorithm and the (reduced) Gröbner basis associated to \mathcal{I} resp. again produce binomials and pure differences.

Lemma 1.33. *Let $<$ be a monomial and \mathcal{I} a binomial.*

- (i) $\text{rem}^1 <; \mathcal{I}$ is a term and $\text{rem}^1 \mathcal{I}; \mathcal{I}$ is a binomial.
- (ii) The reduced Gröbner basis of \mathcal{I} consists of binomials.
- (iii) $\text{NF}^1 <; \mathcal{I}$ is a term and $\text{NF}^1 \mathcal{I}; \mathcal{I}$ is a binomial.

If \mathcal{I} and \mathcal{J} consist of pure binomials, then the preceding statements hold true if “binomial” is replaced with “pure binomial” and “term” by “monomial”.

Proof. (i) If \mathcal{I} is a monomial, then each time a reduction in line 4–7 is applied, the resulting \mathcal{I} is a term again, and even a monomial if \mathcal{I} consists of pure binomials. If no $\text{LT}^1 \mathcal{I} \in \mathcal{I}$, then $A = \mathcal{I}$ is the remainder.

Similarly, if \mathcal{I} is a binomial and $\text{LM}^1 \mathcal{I} \in \mathcal{I}$, then $\mathcal{I} - \frac{\text{LT}^1 \mathcal{I}}{\text{LT}^1 \mathcal{I}} \mathcal{I}$ is again a binomial, as the leading terms in this subtraction cancel. On the other hand, if no $\text{LM}^1 \mathcal{I} \in \mathcal{I}$, then in line 9–10 we have $A - \text{LT}^1 \mathcal{I} = 0$ and $\mathcal{I} - \mathcal{I} \cdot \text{LT}^1 \mathcal{I} = 1 - \mathcal{I}$. From that point on \mathcal{I} is a monomial and the previous discussion applies (with the additional constant factor 1) and the result is a binomial.

Furthermore, if all binomials involved are actually differences of monomials, then the result will again be a pure binomial (note that $1 = 1$ in this case).

(ii) We postpone the proof for a few pages until the introduction of Buchberger’s algorithm, see example 2.7.

(iii) This follows from the previous two statements, as $\text{NF}^1 \mathcal{I}; \mathcal{I} = \text{rem}^1 \mathcal{I}; \mathcal{I}$, where \mathcal{I} is the reduced Gröbner basis of \mathcal{I} .

We now turn to the case of pure binomial ideals and draw a connection to congruence relations on monomials, that is, equivalence relations compatible with multiplication (Definition A.4). We define \sim on $\text{Mon}_=$ if and only if $\alpha - \beta \in \mathcal{I}$. This makes sense for all ideals \mathcal{I} , but for pure difference ideals we obtain a description of the normal form map, Gröbner basis and initial ideal in terms of equivalence classes.

Theorem 1.34 (Kopenhagen & Mayr 1999 [27]). *Let \mathcal{I} be a pure difference ideal with reduced Gröbner basis \mathcal{G} .*

- (i) *For $\alpha \in \text{Mon}_=$, $\text{NF}^{\mathcal{I}}(\alpha)$ is the minimal monomial in the equivalence class $\llbracket \alpha \rrbracket_{\sim}$ (w.r.t. $<$).*
- (ii) *If $\alpha - \beta \in \mathcal{I}$, $\alpha < \beta$, then α is the minimal monomial in $\llbracket \alpha \rrbracket_{\sim}$.*
- (iii) *For any $\alpha \in \text{Mon}_=$ there is a $\beta \in \mathcal{I}$ such that $\alpha - \beta \in \mathcal{I}$.*
- (iv) *$\text{Mon}_= \setminus \mathcal{I}$ consists of the monomials α which are not the minimal element in $\llbracket \alpha \rrbracket_{\sim}$.*

Proof. (i) By the previous Lemma, $\text{NF}^{\mathcal{I}}(\alpha)$ is a monomial which is by definition in $\llbracket \alpha \rrbracket_{\sim}$. We also know that the normal form is a polynomial of minimal multidegree with $\text{LT}(\text{NF}^{\mathcal{I}}(\alpha)) \in \mathcal{I}$ (Lemma 1.23), so $\text{NF}^{\mathcal{I}}(\alpha)$ is the unique minimal monomial in the class of α .

(ii) This is a direct consequence of Theorem 1.32 and statement (i).

(iii) Let $\mathcal{G} = \{g_1, \dots, g_r\}$ be the reduced Gröbner basis of \mathcal{I} , consisting of differences of monomials by the previous lemma. As $\alpha - \beta \in \mathcal{I} = \langle \mathcal{G} \rangle$, there is a $\delta = \sum_{i=1}^r c_i g_i$ with $\text{LT}(\delta) = \alpha - \beta$. Let $\beta = \sum_{i=1}^r b_i g_i$; then $\delta - \beta = \alpha - \beta - \beta \in \mathcal{I}$ has leading term α .

(iv) Let $\alpha \in \text{Mon}_=$, then (iii) yields a β with $\alpha - \beta \in \mathcal{I}$ and $\alpha < \beta$. Conversely, if α is not the minimal element of $\llbracket \alpha \rrbracket_{\sim}$, then by (i) $\text{NF}^{\mathcal{I}}(\alpha) < \alpha$, so $\alpha - \text{NF}^{\mathcal{I}}(\alpha) \in \mathcal{I}$ has leading monomial α .

1.7 Representing polynomials

In order to perform algorithmic manipulation on polynomials we need to fix a way to represent polynomials and their coefficients over a (finite) fixed alphabet. Let $\text{enc}: \mathbb{F} \rightarrow \Sigma^*$ be an encoding, $\text{enc}(0) = \epsilon$. In order to do arithmetic calculations in the field of coefficients, we would like to be able to perform the usual field operations $+$, $-$, \cdot at low computational cost. More precisely, for $0 \leq i < j$, one can calculate $\text{enc}(i \cdot j)$ in polynomial time with

$$\begin{aligned} \text{enc}(2^i) &= \text{enc}(0^i) \cdot \text{enc}(1) & \text{enc}(3^i) &= \text{enc}(0^i) \cdot \text{enc}(4) = \text{enc}(0^i) \cdot \text{enc}(1) \cdot \text{enc}(3) \\ \text{enc}(5^i) &= \text{enc}(0^i) \cdot \text{enc}(1) & \text{if } \text{enc}(0^i) < 0 \end{aligned}$$

Furthermore, equality of elements $\text{enc}^1 0^0 = \text{enc}^1 1^0$ must also be decidable in polynomial time.

This will be our standing assumption for the rest of the thesis. For example, if $\langle \cdot \rangle$ is a finite field, then we can calculate any of the arithmetic operations in constant time with a (large but constant size) lookup table. Infinite fields pose a more interesting challenge.

Example 1.35. If $\langle \cdot \rangle = \mathbb{Z}$ then we can use the following encoding scheme: We encode $a = 0 \cdot 1^2 + \dots$ with $1 \cdot 7^0$, as $\text{bin}^1 0^0 / \text{bin}^1 1^0$. We can then use integer arithmetic to calculate

$$\frac{0}{1} + \frac{2}{3} = \frac{03 + 12}{13} \quad \frac{0}{1} = \frac{0}{1} \quad \dots$$

and can test for equality using $\frac{0}{1} = \frac{2}{3}$ if and only if $03 = 21$. Notice that the denominators will swell pretty quickly, so it might be well-advised to simplify the fraction if desired. y

Remark. Strictly speaking, this is *not* efficient enough for good upper bounds. Instead one should require addition and multiplication to be in AC^0 and NC^1 respectively, in order to obtain a *well-endowed* ring [43, Definition 3.14]. This is necessary in order to enable fast matrix operations as in section 2.5. But we will only consider the integers and rational numbers, which are in fact well-endowed, so we do not elaborate this further.

Suppose now that we have fixed a suitable encoding scheme for $\langle \cdot \rangle$. Then there are two important choices to be made in order to encode polynomials from $\langle \cdot \rangle_{-1}^{\dots} = \mathbb{Z}$:

- (i) Are the exponents of the monomials encoded in binary or is a monomial represented by a sequence of variables? The former is called binary exponent representation, the latter unary representation.
- (ii) Are all coefficients written down (up to the leading term), or only those in the support of f ? The first case is a dense encoding, the second case is sparse encoding.

We will use sparse polynomial encoding with binary exponent representation, although it is remarkable that the lower bounds also hold true with unary exponent notation.

We now explain how to represent and calculate a monomial order. Robbiano showed that any monomial order \prec on $\text{Mon}_{\mathbb{Z}}$ can be represented by a finite set of weight vectors $\gamma_1, \dots, \gamma_n \in \mathbb{Z}^n$, in the following sense: Let $\gamma = (\gamma_1, \dots, \gamma_n) \in \mathbb{Z}^n$, the weight vectors can be interpreted as linear functions

$$\gamma \cdot \alpha = \sum_{i=1}^n \gamma_i \alpha_i$$

Then \prec induces a monomial order with $\alpha \prec \beta$ if and only if there is a $\gamma \in \mathbb{Z}^n$ with

$$\gamma \cdot \alpha < \gamma \cdot \beta \quad \text{and for all } \theta \in \mathbb{Z}^n : \gamma \cdot \theta = 0$$

This is very similar to the definition of the lexicographic order \prec_{lex} , and in a sense *all* monomial orders can be understood as lexicographic orders with weights:

Theorem 1.36 (Robbiano 1985 [44]). *For any monomial order \prec there is a weight matrix $\mathbf{w} \in \mathbb{R}^{n \times n}$ with $\mathbf{w}_{ii} = 1$ and \prec is the graded lexicographic order with respect to \mathbf{w} .*

Example 1.37. In the case of \prec_{lex} the weights have the simple form $w_j = \delta_{j,1}$, so \mathbf{w} is the identity matrix. The graded lexicographic order can be represented by

$$w_j = \begin{cases} 1 & j = 1 \\ 0 & j > 1 \end{cases}$$

Similarly, for \prec_{grevlex} the following weight functions may be used

$$w_j = \begin{cases} 1 & j = 1 \\ -j & j > 1 \end{cases}$$

Dubé, Mishra & Yap [13] gave a constructive proof of this theorem and also showed that the entries of \mathbf{w} may be taken to be non-negative. In the previous example \prec_{grevlex} can be represented by $w_j = \delta_{j,1} - \delta_{j,2}$.

In order to represent an ordering in a finite number of bits, we restrict ourselves to the case where the weights are non-negative rational numbers $w_j \in \mathbb{Q}$. This is not a harsh restriction, as for any \mathbf{w} and $d \in \mathbb{Z}$ there is a rational matrix \mathbf{w}' such that $\prec_{\mathbf{w}'}$ coincides with $\prec_{\mathbf{w}}$ on monomials of degree $\leq d$.

Remark. The representation of monomial orders as weight matrices is mainly useful for theoretical considerations (for example in Chapter 2). In actual computer algebra systems it is usually much faster to directly implement the monomial orders of interest.

Algorithms and upper bounds

In this chapter we describe two fundamentally different approaches to the task of computing Gröbner bases and solving the ideal membership problem. The first is Buchberger's Algorithm, which uses so-called S-polynomials to enlarge a given set of generators to a Gröbner basis. The second approach is based on degree bounds for the polynomials in an ideal membership certificate or a reduced Gröbner basis, and uses linear algebra techniques to obtain exponential space algorithms. We also compare complexity-theoretic upper bounds results for different classes of polynomials.

The material on Buchberger's algorithm is standard to computational commutative algebra texts, we follow [24, Chapter 9]. The content of the later sections is inspired by survey articles of Mayr [32] and Mayr & Toman [35] together with the original literature.

2.1 S-Polynomials

We start with proving a criterion for a set $\{f_1, \dots, f_n\}$ to constitute a Gröbner basis of \mathcal{I} . The greatest common divisor and least common multiple for monomials α, β is defined similarly to the integers as

$$\gcd(\alpha, \beta) = \alpha \beta^{-1} \max(\alpha, \beta) \quad \text{and} \quad \text{lcm}(\alpha, \beta) = \alpha \beta \min(\alpha, \beta)^{-1}.$$

Definition 2.1 (S-Polynomial). Let $0 < \alpha < \beta < \infty$ be two polynomials. The *S-polynomial* is defined as

$$\text{Spoly}(\alpha, \beta) = \frac{\text{LT}(\beta)}{\text{gcd}(\text{LM}(\alpha), \text{LM}(\beta))} \alpha - \frac{\text{LT}(\alpha)}{\text{lcm}(\text{LM}(\alpha), \text{LM}(\beta))} \beta.$$

Equivalently, using $\gcd(\alpha, \beta) = \alpha \beta^{-1} \max(\alpha, \beta)$, we can write

$$\text{Spoly}(\alpha, \beta) = \frac{\text{LC}(\beta) \alpha^0}{\text{LM}(\alpha)} - \frac{\text{LC}(\alpha) \beta^0}{\text{LM}(\beta)}.$$

Example 2.2. The S-polynomial of two monomials is always zero. If α, β are differences of monomials or more generally binomials, then their S-polynomial is also a difference of monomials

resp. a binomial. For example, if $f = x^1 - 1$, $g = x^2 - 2$ with $\delta = 2$, then

$$\text{Spoly}^1 f - g^0 = \frac{-2}{\ell} x^{1-1} - 1^0 - \frac{-1}{\ell} x^{1-2} - 2^0 = \frac{-1 \cdot 2}{\ell} - \frac{-2 \cdot 1}{\ell} = \ell \in \mathbb{B} \text{gcd}(x-1, x^2-2) \cdot y$$

The key property of the S-polynomial is that the leading terms of $\frac{\text{LT}^1 g^0}{\ell} f$ and $\frac{\text{LT}^1 f^0}{\ell} g$ cancel, as both equal $\text{LC}^1 f^0 \text{LC}^1 g^0 \ell^0$. With this notion we can formulate and prove Buchberger’s criterion.

Theorem 2.3 (Buchberger’s criterion). *Let $\mathcal{B} = \{b_1, \dots, b_B\}$ be a finite list of polynomials. The following are equivalent:*

- (a) \mathcal{B} is a Gröbner basis of $\mathcal{B} = \langle \mathcal{B} \rangle$;
- (b) For all $f, g \in \mathcal{B}$ we have $0 \in \text{NF}^{-1} \text{Spoly}^1 f - g^0$;
- (b’) For all $f, g \in \mathcal{B}$ we have $\text{rem}^1 \text{Spoly}^1 f - g^0; \ell^0 = 0$.

Proof. We follow the proof given by Kemper [24, Theorem 9.12].

(a) (b’): (b): All S-polynomials of elements $f, g \in \mathcal{B}$ are linear combinations of elements of \mathcal{B} and hence lie in $\langle \mathcal{B} \rangle$. By characterization 1.18(c) we have $\text{Spoly}^1 f - g^0 \text{rem}^1 \ell^0 = 0$, and thus clearly $0 \in \text{NF}^{-1} \text{Spoly}^1 f - g^0$.

(b) (a): Assume (b) but also that \mathcal{B} is not a Gröbner basis. We will arrive at a contradiction picking a minimal counterexample (with respect to the well-order \prec) and then construct an even smaller counterexample; thus showing that no such counterexample can exist.

Step 1: Construct a minimal expression of a counterexample.

By 1.18(a) there is a $f \in \mathcal{B}$ with $\text{LM}^1 f^0 \notin \langle \text{LM}^1 b_i^0 \rangle$, since $f \in \mathcal{B}$ the set

$$\mathcal{S} = \{x^1 - 1, \dots, x^B - 2\} \prec \langle \mathcal{S} \rangle \quad f = x^1 - 1, \dots, x^B - 2$$

is not empty. Consider the map $\mathcal{S} \rightarrow \mathbb{Z}^B$ which assigns $x^1 - 1, \dots, x^B - 2$ the maximum multidegree $\max_{\delta} \text{mdeg}^1 x^{\delta} b_{\delta}^0$ with respect to \prec . As \prec is a well-order, this map takes its minimal value for some $x^1 - 1, \dots, x^B - 2 \in \mathcal{S}$; let $\ell \in \mathbb{B}$. Since

$$f = \sum_{\delta=1}^B \tilde{c}_{\delta} b_{\delta}^0 \tag{2.1}$$

$\text{LM}^1 f^0$ occurs in some $\text{supp}^1 x^{\delta} b_{\delta}^0$, but not as its leading monomial (as $\text{LM}^1 f^0 \notin \langle \text{LM}^1 b_i^0 \rangle$), so $\text{mdeg}^1 x^{\delta} b_{\delta}^0 < \text{mdeg}^1 f^0$. In particular, ℓ does not occur in $\text{supp}^1 f^0$, and so comparing coefficients of ℓ in (2.1) and using $\text{mdeg}^1 x^{\delta} b_{\delta}^0$ yields

$$0 = \sum_{\delta=1}^B \tilde{c}_{\delta} \text{LC}^1 b_{\delta}^0 - \sum_{\delta=1}^B \tilde{c}_{\delta} \text{LC}^1 x^{\delta} b_{\delta}^0 \quad \text{if } \text{LM}^1 x^{\delta} b_{\delta}^0 = \ell \\ \dots \quad \text{otherwise.} \tag{2.2}$$

to obtain a new representation $f = \sum_{i=1}^n c_i b_i^{\alpha_i}$:

$$f = \sum_{i=1}^n c_i b_i^{\alpha_i} = \sum_{i=1}^n c_i \left(\sum_{j=1}^m z_j b_j^{\beta_j} \right)^{\alpha_i} b_i^{\alpha_i}$$

It remains to show that $\max_i \text{mdeg}_i b_i^{\alpha_i} \leq \max_i \text{mdeg}_i b_i^{\beta_i}$. If $\alpha_i = 0$, then $\text{mdeg}_i b_i^{\alpha_i} = 0$ and hence

$$\text{mdeg}_i b_i^{\alpha_i} = \text{mdeg}_i b_i^{\beta_i} \leq \max_j \text{mdeg}_j b_j^{\beta_j}$$

On the other hand if $\alpha_i > 0$, then $\text{mdeg}_i b_i^{\alpha_i} = \alpha_i \text{mdeg}_i b_i^{\beta_i} = \alpha_i \text{mdeg}_i b_i^{\beta_i} \leq \alpha_i \max_j \text{mdeg}_j b_j^{\beta_j}$ and

$$\text{mdeg}_i b_i^{\alpha_i} \leq \max_j \text{mdeg}_j b_j^{\beta_j} \leq \max_j \text{mdeg}_j b_j^{\beta_j} \leq \max_j \text{mdeg}_j b_j^{\beta_j}$$

Corollary 2.4. Given a set of polynomials $f = \{f_1, \dots, f_n\}$ in $\mathbb{K}[x_1, \dots, x_n]$, the question whether f is a Gröbner basis of $\langle f \rangle$ is decidable.

Proof. By Buchberger's criterion it suffices to verify $\text{rem}(S_{ij}, f) = 0$ for all i, j .

2.2 Buchberger's algorithm

In this section we present Buchberger's algorithm, historically the first algorithm calculating the Gröbner basis of an ideal, introduced by Buchberger in his PhD thesis [8].

Algorithm 4 Buchberger's algorithm

Require: $f = \{f_1, \dots, f_n\}$ in $\mathbb{K}[x_1, \dots, x_n]$

Ensure: B is a Gröbner basis of $\langle f \rangle = \langle f_1, \dots, f_n \rangle$

```

1:  $B \leftarrow \{f_1, \dots, f_n\}$ 
2:  $\% B \leftarrow \{f_1, \dots, f_n\}$ 
3: while  $\% < ;$  do
4:   Pick  $f_i - b^{\alpha_i}$ 
5:    $\% \leftarrow \{f_i - b^{\alpha_i}\}$ 
6:    $B \leftarrow \text{Spoly}(f_i - b^{\alpha_i})$ 
7:    $B \leftarrow \text{rem}(B; \alpha_i)$ 
8:   if  $B < 0$  then
9:      $\% \leftarrow \% \cup \{f_i - b^{\alpha_i}\}$ 
10:     $B \leftarrow \text{Spoly}(\%)$ 
11:   end if
12: end while

```

Theorem 2.5 (Correctness of Buchberger's algorithm). Algorithm 4 terminates after a finite number of steps and calculates a Gröbner basis of the ideal generated by the input f_1, \dots, f_n .

Proof. Termination: We first show that the branch in line 8–10 for $B < 0$ is only taken a finite number of times. Indeed, if $B < 0$, then in particular $\text{LM}^1 B \in \mathfrak{hLM}^1 \mathfrak{o}_i$ and $\text{hLM}^1 \mathfrak{o}_i \subset \text{hLM}^1 \mathfrak{o}_{i+1}$. So each pass of line 10 strictly increases the ideal $\text{hLM}^1 \mathfrak{o}_i$ and by the Noether property of $\langle \rangle_{\mathfrak{K}}$ any ascending chain of ideals must eventually become stationary (see A.1). From this point on each pass through the loop decreases the size of \mathfrak{o} in line 5, so the algorithm terminates with $\mathfrak{o} = \mathfrak{o}_0$.

Correctness: Let \mathfrak{o}_0 be the final result of the algorithm and \mathfrak{o}_i the set at any point in the algorithm. We have the following invariants:

- $\text{hLM}^1 \mathfrak{o}_i = \langle s_1, \dots, s_B \rangle_{\mathfrak{K}}$ (as $s_1, \dots, s_B \in \mathfrak{o}_i$, line 1).
- For $s \in \mathfrak{o}_i$, $B = \text{Spoly}^1 s - \delta^{\circ}$, either $0 \in \text{NF}_{\mathfrak{o}_i} B^{\circ}$ or \mathfrak{o}_i contains a normal form $\text{NF}_{\mathfrak{o}_i} B^{\circ}$ (line 10).

In particular this applies to $\mathfrak{o}_0 = \mathfrak{o}$. If \mathfrak{o} contains a normal form A of $B = \text{Spoly}^1 s - \delta^{\circ}$, then $0 = A - A$ is also a normal form of B with respect to \mathfrak{o} . Hence by Buchberger's criterion 2.3 the set \mathfrak{o} is a Gröbner basis of \mathfrak{o} .

Example 2.6. We take the polynomials $\delta_1 = x^2 - 1$, $\delta_2 = x^2 - 1$ from example 1.16 with $\mathfrak{K} = \mathbb{C}[x]$. We have $B_1 = \text{Spoly}^1 \delta_1 - \delta_2^{\circ} = 0$ and $B_1 = \text{rem}^1 B; \delta_1 - \delta_2^{\circ} = B_1$. Hence we add B_1 to \mathfrak{o} and consider the new pairs $B_1 - \delta_1^{\circ}$ and $B_1 - \delta_2^{\circ}$. We have

$$B_2 = \text{Spoly}^1 B_1 - \delta_1^{\circ} = x^2 - 1 - (x^2 - 1) = 0 \quad B_3 = \text{Spoly}^1 B_1 - \delta_2^{\circ} = x^2 - 1 - (x^2 - 1) = 0$$

Now $B_2 = \text{rem}^1 B_2; \delta_1 - \delta_2 - B_1^{\circ} = 0$ and $B_3 = \text{rem}^1 B_3; \delta_1 - \delta_2 - B_1^{\circ} = 0$, hence the algorithm terminates and we obtain the Gröbner basis $\{x^2 - 1\}$.

Remark. Consider a set of polynomials $\{s_1, \dots, s_B\} \subset \mathfrak{K}[x]$ containing $x^2 - \delta$ for $\delta \in \mathfrak{K}$. These polynomials enforce that any solution to this system takes values in $\mathfrak{K}[x]$. In this way Boolean formulae and equations can be expressed using polynomial systems, and one can techniques from commutative algebra to reason about such statements. For an interesting example see the "Gröbner proof system" by Clegg, Edmonds & Impagliazzo [10], who use Gröbner bases and Buchberger's algorithm to produce proof certificates for tautologies.

If one is interested in a reduced Gröbner basis, then algorithm 5 can be applied after producing an arbitrary Gröbner basis \mathfrak{o} .

The correctness of this algorithm follows from the results and proofs of Lemma 1.25 and Theorem 1.28. For example, the Gröbner basis from the previous example can be reduced to $\{x^2 - 1\}$. As a first application, we fill the gap in the proof of lemma 1.33 as promised.

Example 2.7 (The reduced Gröbner basis of a binomial ideal). If $\mathfrak{o} = \langle s_1, \dots, s_B \rangle_{\mathfrak{K}}$ is a set of binomials, then any S-polynomial B formed from this set is also a binomial, and so is $B \bmod \mathfrak{o}$ by Lemma 1.33(i). Hence Buchberger's algorithm adds only binomials to \mathfrak{o} and thus produces some Gröbner basis consisting of differences of monomials. Similarly, the algorithm reducing

Algorithm 5 Reduction of a Gröbner basis**Require:** G a Gröbner basis of I **Ensure:** G^0 is the unique reduced Gröbner basis of I

```

1:  $G^0 \leftarrow G$ ;
2: for all  $g \in G^0$  do
3:   if  $g \in I$  and  $\text{LM}(g) \in \text{LM}(G^0)$  for all  $g \in G^0$  then
4:      $g \leftarrow \text{LC}(g) \cdot \text{LC}^{-1}(\text{LM}(g))g$ 
5:   end if
6: end for
7:  $G^0 \leftarrow G^0$ ;
8: for all  $\delta = 1 \dots j$  do
9:    $\text{rem}(g_\delta; G^0) \leftarrow [f \delta_{\delta_1} \dots \delta_{\delta_j}]$ 
10:   $G^0 \leftarrow G^0 \cup [f \delta]$ 
11: end for

```

▶ Interreduction
 ▶ Normalization
 ▶ $G^0 = \{g_1, \dots, g_r\}$
 ▶ Reduction

this Gröbner basis will only delete or normalize elements from G and then introduce some remainders which are again binomials. Hence the reduced Gröbner basis consists of binomials. The same argument works for pure binomials and for homogeneous polynomials. \square

We also note that the combination of Buchberger's algorithm and the normal form algorithm yield:

Theorem 2.8. *The ideal membership problem IM_\prec and (reduced) Gröbner bases GROEBM_\prec are decidable (as long as the field operations are computable).*

The run-time of Buchberger's algorithm depends heavily on the number of Polynomials B which have to be added to G . Also, the majority of the computation is spent calculating the remainder of the B with respect to increasingly complicated sets G . A good implementation uses various techniques to reduce the number of unnecessary normal form calculations, for example [19, Exercise 1.7.2]:

Lemma 2.9 (Product criterion). *If $f = \delta_1 \delta_2 \dots \delta_j$ with $\text{gcd}(\delta_1, \dots, \delta_j) = 1$, then*

$$0 \in \text{NF}_{f \delta_1 \dots \delta_j}(\text{Spoly}(f, \delta_1, \dots, \delta_j))$$

and hence $\text{Spoly}(f, \delta_1, \dots, \delta_j)$ may be skipped in Buchberger's algorithm.

Remark. One of the most widely used improvements of Buchberger's algorithm are the algorithms 4 and 5 by Faugère [16]. For some bounds on the complexity of 5 see for example the work of Bardet, Faugère & Salvy [2].

2.3 Degree bounds

In this section we gather some upper bounds on the degrees of polynomials appearing in the problems of our interest, following Mayr [32] and Mayr & Toman [35]. We start with the

ideal membership problem IM_k , let $f = f_1 \cdots f_B \in \mathbb{A}^n$, then we are interested in bounds depending on the following factors:

- B , the number of generators $f_1 \cdots f_B$.
- n , the number of variables in $\mathbb{A}^n = \mathbb{F}[x_1, \dots, x_n]$.
- $d = \max\{\deg f_1, \dots, \deg f_B\}$, the degrees. Without loss of generality we may assume $d = \deg f_1$ and $d \geq \max_{2 \leq i \leq B} \deg f_i$.

In order to decide whether $f \in \langle f_1, \dots, f_B \rangle$, one could try to systematically search the space of polynomials \mathbb{A}^n in order to find a solution to

$$f = g_1 f_1 + \dots + g_B f_B \quad (2.3)$$

A first step in order to make this feasible is to bound the degree of the g_i , historically the first result is due to Hermann, see Mayr & Meyer for a short proof [33, Appendix].

Theorem 2.10 (Hermann 1926 [20]). *If the equation (2.3) admits a solution, then there is a solution with*

$$\deg g_i \leq 3^{B-1} d^{2^{B-1}} \quad \forall i = 1, \dots, B.$$

In the next section we will see how this and the bounds below may be used to turn ideal membership, normal form and Gröbner basis calculation into problems from linear algebra.

A natural question is to ask whether this degree can be improved, since a better bound yields a system of linear equations with fewer variables and equations. In this generality, the double-exponential nature of the Hermann bound is unavoidable in the following sense: For any $n \geq 3$ there exist polynomials $f = f_1 \cdots f_B$, where $B = 10$, the number of variables is $n = 10$ and the degree is $O(3^n)$, such that any solution to (2.3) has at least one g_i with

$$\deg g_i \geq 3^{2^{B-1}}.$$

This result is due to Mayr & Meyer [33], we will prove a variant of this in chapter 3. On the other hand, if the f_i satisfy additional properties, then better bounds can be proved. The case of homogeneous polynomials has already been mentioned in Lemma 1.7, there the degrees of the g_i are simply bounded by d .

Another important special case is the case $f = 1$, i. e. the Nullstellensatz case for HNST_k . Here a single exponential bound was proven by Brownawell for $k = \mathbb{C}$, which was improved and extended to arbitrary fields by Kollár.

Theorem 2.11 (Kollár 1988 [26]). *If $f = 1 \in \langle f_1, \dots, f_B \rangle$, then there exist g_1, \dots, g_B with $1 = \sum_{i=1}^B g_i f_i$ such that*

$$\deg g_i \leq \deg f_i + \max_{1 \leq j \leq B} \{3 - 3g_j\} \quad \forall i = 1, \dots, B.$$

This result is sometimes called the *effective Nullstellensatz*, as it yields explicit bounds on the degrees occurring in Hilbert’s Nullstellensatz.

Finally, we give an example on how the dimension of the ideal influences the degree bound, for more on dimension see for example [19, Chapter 5].

Definition 2.12 (Dimension of an ideal). The *dimension* $\dim^1 \mathfrak{a}$ of an ideal $\mathfrak{a} \subseteq \mathbb{K}[x_1, \dots, x_n]$ is the largest integer $d \leq n$ such that there exist variables $x_1, \dots, x_d \in \mathbb{K}[x_1, \dots, x_n]$ with

$$\mathbb{K}[x_1, \dots, x_d] \cap \mathfrak{a} = \{0\}.$$

Theorem 2.13 (Dickstein et al. 1991 [12]). Assume (2.3) admits a solution and the ideal \mathfrak{a} is of dimension zero. Then there is a solution with

$$\deg^1 \delta \leq 3^{2^n} \cdot 3^n \cdot \deg^1 f.$$

We now turn to degree bounds of Gröbner bases, let \prec be a monomial order on Mon_n . We can reduce to the homogeneous case as follows: Let $\mathfrak{a} = \langle f_1, \dots, f_r \rangle$ and add a new variable t .

Definition 2.14 ((De)homogenization). Let $\mathfrak{a} \subseteq \mathbb{K}[x_1, \dots, x_n]$ of degree d , then its *homogenization* is

$$h_{\mathfrak{a}} = \left\langle \sum_{j=0}^d f_j \frac{x_1^j \dots x_n^j}{t^{d-j}} \mid f_j \in \mathfrak{a} \text{ of degree } j \right\rangle \subseteq \mathbb{K}[x_1, \dots, x_n, t].$$

If $\mathfrak{a} \subseteq \mathbb{K}[x_1, \dots, x_n]$ is homogeneous, then its *dehomogenization* is

$$a_{\mathfrak{a}} = \mathfrak{a} \cap \mathbb{K}[x_1, \dots, x_n] \subseteq \mathbb{K}[x_1, \dots, x_n].$$

We define an ordering on $\text{Mon}^1 \mathbb{K}[x_1, \dots, x_n, t]$ via \prec_h if and only if

$$\deg - 5 \deg - \text{ or } \deg - = \deg - \text{ and } x_1^{-1} \dots x_n^{-1} t^{-1} \prec_h \cdot$$

Lemma 2.15. Let $\mathfrak{a} = \langle f_1, \dots, f_r \rangle \subseteq \mathbb{K}[x_1, \dots, x_n]$ be an ideal and $h_{\mathfrak{a}} = \langle h_{f_1}, \dots, h_{f_r} \rangle \subseteq \mathbb{K}[x_1, \dots, x_n, t]$. If g_1, \dots, g_r is a homogeneous Gröbner basis of $h_{\mathfrak{a}}$ with respect to \prec_h , then $a_{\mathfrak{a}} = \langle g_1, \dots, g_r \rangle$ is a Gröbner basis of \mathfrak{a} .

This is not a difficult result, but it requires additional concepts from commutative algebra, see for example [37, Lemma 2.5] or [34, Lemma 7]. This shows that upper bound on the degree of Gröbner bases elements of homogeneous ideals (in $n + 1$ variables) automatically yields bounds on the degree of Gröbner bases of arbitrary ideals (in n variables). Some conditional results in this direction were obtained by Möller & Mora [37], but the most influential result

is the unconditional degree bound by Dubé:

Theorem 2.16 (Dubé 1990 [14]). Let $I = \langle f_1, \dots, f_B \rangle \subseteq \mathbb{K}[x_1, \dots, x_n]$ be an ideal and $3 = \max_{\delta} \deg f_{\delta}$. Then the reduced Gröbner basis of I consists of polynomials g_{δ} of degree

$$\deg g_{\delta} \leq 2 \frac{3^2}{2} \cdot 3^{2^{\delta-1}}.$$

If the f_{δ} are homogeneous, then the exponent can be improved to $2^{\delta-2}$.

The proof uses the Hilbert function of a homogeneous ideal and cone decompositions, for an overview see the paper by Mayr & Ritscher [34], who also prove dimension-dependent upper bounds.

Theorem 2.17 (Mayr & Ritscher 2013 [34]). Let \mathbb{K} be an infinite field. Let $I = \langle f_1, \dots, f_B \rangle \subseteq \mathbb{K}[x_1, \dots, x_n]$ be an ideal of dimension A generated by homogeneous f_{δ} of degrees $3_1, \dots, 3_B$. Then the reduced Gröbner basis of I consists of polynomials g_{δ} of degree

$$\deg g_{\delta} \leq 2 \frac{1}{2} 3_1 \cdot 3_{=A} \cdot 3_1^{2^{\delta-1}}$$

If the polynomials are not necessarily homogeneous, then a similar bound with exponent 2^{δ} can be achieved. The degree bound by Dubé will enable us to prove an exponential space upper bound on the complexity of computing Gröbner bases.

2.4 From polynomials to linear algebra

We continue to use notation from the previous section with $f = f_1, \dots, f_B$ of degrees $3^0 = 3_1, \dots, 3_B$, $3 = \max_{\delta} 3_{\delta}$ in n variables. Furthermore, let \prec be a monomial order, given by a weight matrix $W \in \text{Mat}^1 = \mathbb{R}^{n \times n}$ as in section 1.7. We will now describe how to translate the following problems into problems of linear algebra:

- (i) Decide whether $f = \sum_{\delta=1}^B c_{\delta} f_{\delta}$ for suitable $c_{\delta} \in \mathbb{K}$.
- (ii) Calculate $\text{NF}(f)$, where $f = f_1, \dots, f_B$ with respect to \prec .

In order to obtain concrete upper bounds on the space required by a Turing machine to solve these problems, we work over the field of rational numbers $\mathbb{K} = \mathbb{Q}$ from here on until the end of this chapter. The input to these algorithms consists of the sequence of polynomials $f = f_1, \dots, f_B$ and in second case the weight matrix W . The rational numbers and the exponents of the monomials are encoded in binary (or fractions of binary numbers). Let n be the length of the input, then we have the following estimates

$$3^0 = 3 \cdot 2^{-n} = -B \cdot \dots \tag{2.4}$$

We first consider the case of ideal membership. Let δ be a estimate of the degree of a set of solution $\deg \delta$, for example the Hermann bound $\delta = 3^0 \dots 1B3^{o2}$. We use the notation f_j to indicate the coefficient of x_j in f . Then the solution of (i) depends on the finitely many coefficients in

$$f = \sum_{2Z=j} \tilde{O} \delta_j \dots \delta_j \cdot$$

A standard counting argument shows that the number of monomials of degree δ in n variables is $\binom{n+\delta}{\delta} = \binom{n+\delta}{n}$. Finding a solution to (i) is then equivalent to solving the following enormous system of linear equations:

- **Variables:** $\tilde{O} B \tilde{B} = \dots$ unknowns δ_j , indexed by $f_1 \dots Bg \ f \ 2 \ \tilde{Z} = j \ j \ j \ g$.
- **Equations:** $A B = \dots$ equations where $\delta = 3 \dots$, corresponding to the coefficient of x_j in (i)

$$f = \sum_{2Z=j} \tilde{O} \tilde{B} \delta_j \dots \delta_j \cdot$$

By construction the ideal membership problem is equivalent to solvability of this system of linear equations. In order to represent this as a matrix, we choose an ordering of the sets $\{x_j\}$ of monomials of degree δ and δ^0 respectively, for example lex . Then it makes sense to talk about the δ -th monomial in each set, and the system of equations can be written as

$$A = B \dots A \ 2 \ Mat^1 A \ @-' \ ^0 \ 2' \ @ \ B \ 2' \ A.$$

In this presentation we have:

- B_δ is the coefficient of the δ -th monomial in f (which is zero if $\delta > 3^0$).
- Write $\delta = \sum_{j=1}^n \delta_j x_j$, then δ_j is the coefficient of the δ_j -th monomial in \dots
- $A_{\delta, \delta}$ consists of the coefficients of the f : as follows: Let x_j be the δ -th monomial x_j be the δ_j -th monomial in f : (as in the previous case), then $A_{\delta, \delta} = f$.

In the language of linear algebra, the equation $A = B$ has a solution if and only if B is in the linear span of the columns of A , or equivalently if the matrix $\llbracket A \ j \ B \rrbracket$ has the same rank as A . This reduces the ideal membership to the task of computing a rank of a matrix. The size of this matrix can be estimated in the input length as follows:

- The entries of A and A^0 are coefficients of $f = f_1 \dots f_B$, and hence of size \dots .
- The degree δ is bounded by

$$\delta = 3^0 \dots 1B3^{o2} \stackrel{(2.4)}{\leq} \sum_{j=1}^n 2^{o2} \ 2^{2^3}$$

using the generous estimate ≤ 2 .

- The number of equations A is bounded by

$$A = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} 1 = \sum_{i=0}^{\infty} (2^{i+1} - 1) = \sum_{i=0}^{\infty} 2^{i+1} - \sum_{i=0}^{\infty} 1 = 2^{2^5} - 2^{2^5}.$$

- Similarly, the number of variables $@$ is bounded by

$$@ = B = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} 1 = \sum_{i=0}^{\infty} (2^{i+1} - 1) = \sum_{i=0}^{\infty} 2^{i+1} - \sum_{i=0}^{\infty} 1 = 2^{2^5} - 2^{2^5}.$$

Thus we have proven

Theorem 2.18. *The ideal membership problem $IM_{\mathbb{C}}$ can be reduced to the rank computation of two matrices of size $2^{2^5} \times 2^{2^5}$ whose entries are coefficients of the input. Furthermore, the polynomials g_j can be obtained as solutions to a system of linear equations of the same characteristics.*

We now turn to the calculation of the normal form. Here a similar approach may be used, namely we have the polynomial equation

$$0 = NF^{-1}f^0 = \sum_{\beta=1}^{\infty} \sum_{\gamma=1}^{\infty} \sum_{\delta=1}^{\infty} f_{\beta\gamma\delta} g_{\beta\gamma\delta} \tag{2.5}$$

and we are looking for a solution with $\text{supp } f^0$ minimal (Lemma 1.23). In order to do this we need an estimate on $\text{deg } NF^{-1}f^0$. Let

$$= \max_{\beta, \gamma, \delta} \{ \sum_{i=1}^{\beta} w_i - \sum_{j=1}^{\gamma} w_j - \sum_{k=1}^{\delta} w_k \} \text{ in } g$$

be the largest natural number occurring in the weight matrix w , and let B be a bound on the degree of the reduced Gröbner basis of $\mathbb{C}[x_1, \dots, x_n]$. Kühnle & Mayr [30, Section 2], expanding on [13], proved the following upper bound:

Theorem 2.19. *For any $f \in \mathbb{C}[x_1, \dots, x_n]$ we have*

$$\text{deg } NF^{-1}f^0 \leq \sum_{i=1}^n w_i + 2 \cdot \text{deg } f^0 + B \cdot \sum_{i=1}^n w_i.$$

Remark. If a specific monomial order is considered, then better bounds are available. For example, if w is degree-dominated, i.e. $\text{deg } w_i \geq 7 \text{ deg } w_j$ implies $w_i \geq w_j$, then of course $\text{mdeg } NF^{-1}f^0 \leq \text{mdeg } f$ implies $\text{deg } NF^{-1}f^0 \leq \text{deg } f$.

We applying the Hermann bound to $f = NF^{-1}f^0 = \sum_{\beta=1}^{\infty} \sum_{\gamma=1}^{\infty} \sum_{\delta=1}^{\infty} f_{\beta\gamma\delta} g_{\beta\gamma\delta}$ to obtain

$$\text{deg } g_{\beta\gamma\delta} \leq \text{deg } f + \text{deg } g_{\beta\gamma\delta} \leq \sum_{i=1}^n w_i + B \cdot \sum_{i=1}^n w_i + \sum_{i=1}^n w_i \in \mathbb{C} \cdot i.$$

We use a strategy similar to the ideal membership case to turn (2.5) into a system of linear

equations, for additional details see [43, Chapter 6]. We introduce an extra factor $f_0 = 1$ to recreate the shape of the ideal membership equation

$$f_0 \cdot f_1 \cdot \dots \cdot f_B = \sum_{\alpha} A_{\alpha} \cdot B_{\alpha}$$

This time α contains the coefficients of $x_1 \cdots x_B$, β contains the coefficients of x_0 , A contains the coefficients of $f_1 \cdots f_B$, α contains 0's and 1's for the coefficients of f_0 and B contains the coefficients of f .

We now sketch how this system can be used to find the normal form $f = NF(f)$. We know that there is *some* solution (for example $x_0 = f$) when all monomials in α are allowed. The normal form is characterized as the unique polynomial $f \in \mathbb{F}[x_0, \dots, x_B]$ with minimal support $\text{supp}(f)$ with respect to \prec (see Lemma 1.23). The key idea is to systematically remove monomials from α together with the corresponding row from A and see whether the system still has a solution. We order the columns of A so that the columns in α are in ascending order with respect to the monomials in α , i.e. the lowest monomials $x_1 \cdots x_B$ come first. Then we have a maximal regular minor (an invertible square sub-matrix of maximal size) corresponding to the minimal solution α . One can decide whether a given column indexed by i is contained in this minor (i.e. if the corresponding monomial x^{α_i} is in $\text{supp}(f)$) by comparing the rank of the minor of the first i rows/columns to the rank of the first $i-1$ rows/columns: If the rank is different, then x^{α_i} is indispensable, otherwise it is not in the support.

The maximal degree of a monomial in (2.5) is bounded by $\max_i \sum_{j=1}^B g_j = \sum_{j=1}^B g_j$, so the number of equations is bounded by $\sum_{i=1}^B \sum_{j=1}^B g_j = 10^6$. Using similar estimates as before, we obtain the following result.

Theorem 2.20. *The computation of (elements of) the support $\text{supp}(NF_{\{f_1, \dots, f_B\}}(f))$ can be reduced to rank computations of matrices of size $2^{2^{O(g_1)}} \times 2^{2^{O(g_1)}}$ whose entries are coefficients of the input (or 0-1). Furthermore, the normal form itself can be obtained as solutions to a system of linear equations of the same characteristics.*

We note here that this is true even if the degree estimate for $\deg(f)$ is not 2 but rather $2^{2^{O(g_1)}}$, this will become important when enumerating a Gröbner basis in algorithm 6.

2.5 Fast linear algebra on PRAMs

In this section we briefly introduce the PRAM model for parallel computation, for details and a modern treatment see for example the book by Parhami [40]. Since we will only use abstract results about the complexity on PRAMs, only an overview of the model is given.

A PRAM (parallel random access machine) consists of

- a set of global registers, containing arbitrary integers,
- a number of processors p_0, \dots, p_{p-1} (depending on the input size), each with their own local registers,
- a finite program consisting of arithmetic operations, branching and read/write access to local or global memory.

Each processor runs on the same program with the knowledge of its id $0 \leq i < p$ and the number p . The input is contained in the global register, for example if the input is a matrix $A \in \mathbb{R}^{n \times n}$, then the global registers contain the A_{ij} coefficients. The execution is performed synchronously, and read/write conflicts are handled “properly”.

The efficiency of an algorithm is expressed both in the number p of processors used, as well as the number of steps T taken until each processor halts. A central result relating the PRAM model to (deterministic) Turing machines is the *Parallel Computation Thesis*.

Theorem 2.21 (Fortune & Wyllie 1978 [17]). Let $T = O(\log^c n)$.

- (i) If $L \in \text{SPACE}^c$, then L is accepted by a PRAM in parallel time $O(T)$.
- (ii) If L is accepted by a PRAM in parallel time T , then $L \in \text{SPACE}^{O(T)}$.

In particular, parallel time complexity is polynomially related to sequential space complexity.

Thus, in order to obtain space-efficient algorithms for problems such as rank computation, we can employ efficient parallel algorithms. Of particular relevance are the following results due to Csanky.

Theorem 2.22 (Csanky 1976 [11]). Given an integer $n \times n$ matrix, the tasks of

- (i) matrix inversion
- (ii) solving a system of linear equations
- (iii) computing the determinant
- (iv) computing the coefficients of the characteristic polynomial

can be solved in parallel time $O(\log^2 n)$ using $n^{O(1)}$ processors.

Ibarra, Moran & Rosier extended this list to include matrix rank calculation.

Theorem 2.23 (Ibarra, Moran & Rosier 1980 [22]). The problem of calculating the rank of an $n \times n$ matrix can be solved in parallel time $O(\log^2 n)$ using $n^{O(1)}$ processors.

Remark. The papers cited here use the arithmetic machine model, i. e. directly manipulating integers (or rational numbers). Some care has to be taken in order to translate these results to the bit model, Pan describes some appropriate methods [39].

2.6 Upper bounds on ideal membership

We finally apply the space-efficient linear algorithms from the previous section to the ideal membership and normal form problems.

Theorem 2.24 (Mayr 1989 [31], IM, ≤ 2 EXPSPACE). *There exists a deterministic algorithm which on input f_1, \dots, f_B decides whether there exist x_1, \dots, x_B with $f = \sum_{i=1}^B x_i f_i$, using no more than exponential working space.*

Proof sketch. Using polynomial space we can clear denominators of the input polynomials f_1, \dots, f_B and hence may assume that all polynomials have integer coefficients. We first describe a *parallel* algorithm for this task.

Theorem 2.23 describes an algorithm for matrix rank calculation in parallel time $O(\log n)$, where n is the size of the matrix. The matrices A, A^0 from Theorem 2.18 have size $2^{O(n)}$ and hence there is a parallel algorithm calculating the rank of A in time $2^{O(n)}$ using $2^{O(n)}$ processors. Notice that we cannot write down these matrices explicitly (in exponential space), so instead each time this parallel algorithm requests an entry of A (by indexing it), we calculate it “on the fly” from the input polynomials. The time required to do this is negligible, since the entries are coefficients of the f_i which can be indexed efficiently.

Now apply the parallel computation hypothesis (Theorem 2.21) to turn this exponential parallel time algorithm into an exponential space algorithm.

With some additional care one can extend this algorithm to output the solution in the same space bounds, for details see Theorem 4 in the work of Mayr [31].

Theorem 2.25. *With notation as in the previous theorem, if a solution exists, then the algorithm can write it to an output tape in exponential work space.*

Notice however that the size of the x_i themselves may be double-exponential by the Hermann bound and the results of chapter 3, hence it is important to distinguish the work tape from the output tape. A similar argument proves the analogous statement for the normal form, see also [43, Theorem 6.2].

Theorem 2.26. *There is an exponential space algorithm which on input f_1, \dots, f_B computes the normal form $NF_{\{f_1, \dots, f_B\}}(f)$.*

Again, the size of the normal form may exceed exponential space, so care has to be taken if this algorithm is used as a subroutine.

If the polynomials are of a specific type, then we can use the other bounds from section 2.3 to give better estimates on the degrees of the x_i . This leads to matrices of single exponential space, and we conclude:

Theorem 2.27 (IM_h, \leq HNST, ≤ 2 PSPACE). *In the situation of Theorem 2.24, if*

- (i) *the polynomials f_1, \dots, f_B are homogeneous or*

(ii) $\delta = 1$ (the “Nullstellensatz” case),

then the corresponding problem can be solved in polynomial space.

While the homogeneous ideal membership problem $\text{IM}_{h,\delta}$ is indeed hard for PSPACE (and hence complete), for the problem HNST_δ much better results are available:

Theorem 2.28 (Koiran 1996 [25]). *Under the assumption of the Generalized Riemann Hypothesis (GRH) we have $\text{HNST}_\delta \leq 2 \text{RP}^{\text{NP}} \leq \text{PH}$, in fact, $\text{HNST}_\delta \leq 2 \text{AM} = \text{BP} \leq \text{NP}$.*

For the definition the Arthur-Merlin class see for example [1, Section 8.2]. Since HNST_δ is NP-hard, this implies the surprising result that (under the GRH assumption) $\text{P} < \text{NP}$ if and only if $\text{HNST}_\delta \leq \text{P}$. For details and improvements (weakening the GRH assumption) see the work of Rojas [45]. An overview over known complexity bounds for several variants of ideal membership is presented by Mayr & Toman [35, Table 1].

2.7 Computing a Gröbner basis in EXPSPACE

We finally present an algorithm computing the reduced Gröbner basis of an ideal $I = \langle f_1, \dots, f_m \rangle$ in exponential space. It is important to distinguish the work tape from the output tape, since the output may consist of double-exponentially many polynomials, each of which might contain double-exponentially many terms. The key to algorithm 6 is to enumerate the possible leading monomials one by one, and then comparing them to their normal forms in a sufficiently space-efficient way.

Algorithm 6 Enumerating the reduced Gröbner basis

Require: $f_1, \dots, f_m \in \mathbb{K}[x_1, \dots, x_n]$, \mathcal{M} , a weight matrix for

Ensure: Yields all elements of the reduced Gröbner basis of $I = \langle f_1, \dots, f_m \rangle$

```

1:  $\mathcal{M} \leftarrow \max_{f \in \{f_1, \dots, f_m\}} \text{deg } f$ 
2:  $\mathcal{M} \leftarrow \mathcal{M} \oplus \mathcal{M}^{\oplus 2} \oplus \dots \oplus \mathcal{M}^{\oplus 2^{\mathcal{M}}}$  ▷ The Dubé bound 2.16
3: for all  $\tilde{z} = (z_1, \dots, z_n) \in \mathcal{M}$  do
4:    $\tilde{m} \leftarrow \tilde{z}$ 
5:   if  $\tilde{m} \in \text{NF}_{h_{f_1, \dots, f_m}}^{\mathcal{M}} \leftarrow \mathcal{O}$  then
6:     continue in line 3 ▷  $m$  is not reducible
7:   end if
8:   for  $\theta \in \{1, \dots, \mathcal{M}\}$  do
9:      $\tilde{m} \leftarrow \tilde{m} \oplus \theta \cdot \mathcal{M}^{\oplus \theta}$ 
10:    if  $\tilde{m} \in \text{NF}_{h_{f_1, \dots, f_m}}^{\mathcal{M}} \leftarrow \mathcal{O}^{\oplus \theta}$  then
11:      continue in line 3 ▷  $m$  is not minimally reducible
12:    end if
13:  end for
14:  yield  $\tilde{m} \in \text{NF}_{h_{f_1, \dots, f_m}}^{\mathcal{M}} \leftarrow \mathcal{O}$  ▷  $m$  is minimally reducible
15: end for

```

Theorem 2.29. *Algorithm 6 enumerates the reduced Gröbner basis of $\mathcal{B} = \{f_1, \dots, f_B\}$ in exponential work space and double-exponential time.*

Proof. Correctness: By Theorem 1.32, the reduced Gröbner basis consists of polynomials of the form $\langle \text{NF}_{\mathcal{B}} \langle \cdot \rangle$ where \langle is a monomial which is minimally reducible. The Dubé bound gives an upper bound on the degree of the elements of \mathcal{B} (Theorem 2.16), i. e. of \langle . The algorithm iterates through all possible leading terms \langle (line 3) and checks if \langle is reducible (line 5-8) and if any divisor $\langle^0 \cdot j \langle$ is irreducible (line 11-14). It suffices to check this for \langle^0 with $\deg \langle^0 = \deg \langle - 1$, so in line 16 the algorithm has verified that \langle is minimally reducible and it yields the corresponding element.

Space requirement: Let n be the input size, then $B \leq n$ and $3 \leq n$. With this we have $2^{2^{O(n)}}$, which fits into the working memory of exponential size.

We can enumerate the monomials \langle of degree n in any order we like. In order to do this in exponential space, use n binary counters $\langle_1, \dots, \langle_n$ from 0 to n and then take $\langle = \langle_1^{1} \dots \langle_n^{n}$. This requires $O(n) = \log n$ space.

We use the normal form algorithm from the previous section on input $\langle \in \mathcal{B}^*$; this requires exponential space in \mathcal{B}^* (!) by the remark after Theorem 2.20. We can not afford to write down the resulting polynomial in its entirety, but we can check (term by term) whether it coincides with \langle . This technique is applied in line 5 and 10, while in line 14 we can write the result (with an additional minus sign) to the output tape.

Time requirement: The number of configurations of an algorithm working in space $2^{O(n)}$ is bounded by $2^{2^{O(n)}}$, and hence its time requirement is at most double-exponential.

Corollary 2.30 (GROEBM \in EXPSPACE). *There is an exponential space algorithm deciding whether δ is contained in the reduced Gröbner basis of $\mathcal{B} = \{f_1, \dots, f_B\}$.*

Proof. Indeed, modify algorithm 6 as follows:

- In line 15, instead of writing $\delta \in \text{NF}_{\mathcal{B}} \langle$ to the output, compare it to δ , if they coincide then accept.
- After line 17, reject.

This algorithm has essentially the same space requirements and decides GROEBM.

Remark. This algorithm is worst-case optimal in the sense that GROEBM is EXPSPACE-hard and there exist inputs \mathcal{B} such that the corresponding reduced Gröbner basis indeed has double-exponentially many elements. On the other hand, this algorithm is “uniformly impractical” in the sense that even if \mathcal{B} consists of few elements, then algorithm 6 still requires exponential space (as n is extremely large, and the normal form algorithm also uses exponential space on all inputs).

Ritscher improved this approach using S-polynomials and by increasing n only as far as necessary to obtain a more space efficient algorithm [43]. For the class of pure binomial

ideals Koppenhagen & Mayr [27] devised an algorithm using combinatorial tools. While this algorithm requires exponential space (this is necessary, see chapter 3), it does not make use of the parallel computation hypothesis.

For low-dimensional ideals better bounds are known.

Theorem 2.31 (Krick & Logar 1991 [29]). *If $I = \langle f_1, \dots, f_b \rangle \subseteq \mathbb{C}[x_1, \dots, x_n]$ is an ideal of dimension $d \leq 1$, then a Gröbner basis can be calculated in time $2^{O(n^d)}$ assuming unit cost arithmetic of \mathbb{C} .*

Lower bounds

In this chapter we prove an exponential space lower bound for the ideal membership problem and the reduced Gröbner basis membership problem. For this we introduce the notion of Thue systems, which are essentially presentations of semigroups. The key result is that there is a family of commutative Thue systems which can produce words of double exponential length in the size of the presentation. The chain of reductions from a generic EXPSPACE-problem to Gröbner bases is displayed in Figure 3.1. We also introduce the concept of Church-Rosser systems, which are a string-replacement analogue of Gröbner bases. These notions will lead to examples of ideals with Gröbner bases of double-exponential length and degree.

The primary source for most of the chapter is the original work by Mayr & Meyer [33] and Huynh [21].

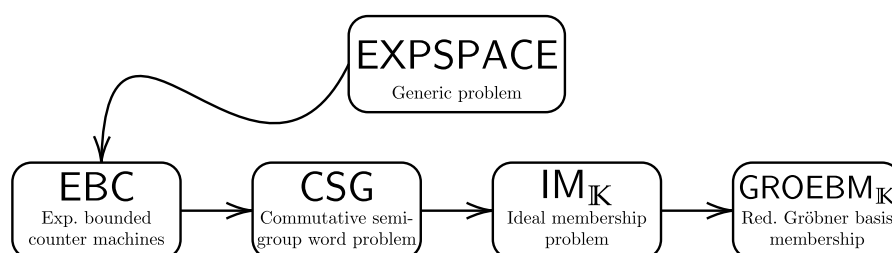


Figure 3.1: The chain of complexity-theoretic reductions.

Remark. In the literature, for example [33] and [32], the type of reduction is not polynomial time many-one reductions. Instead, more restrictive log-lin reductions are used: The reduction function f may use logarithmic space and $|f^1G^0| \leq O^1|G^0|$. With this notion, the problems considered here are not complete for $\text{EXPSPACE} = \bigcup_{c \geq 1} \text{SPACE}^{O^1=^c}$, but rather $\text{SPACE} \leq \text{SPACE}^{O^1=^c}$. This has the advantage that if π is a SPACE -hard problem with respect to log-lin reductions, then there is a $c \geq 0$ such that $\pi \in \text{SPACE}^{O^1=^c}$, proving exponential resource lower bounds [33, Section 2] (this is essentially an application of the space hierarchy theorem [1, Theorem 4.8]).

In this thesis we only consider polynomial time reductions for simplicity, although the interested reader will have no difficulty in verifying that the reductions $\text{EBC} \leq \text{CSG} \leq \text{IM}_\mathbb{K} \leq \text{GROEBM}_\mathbb{K}$ are actually log-lin reductions.

translation still works if the reverse rules are included, so string equivalence with respect to a Thue system is also undecidable (in fact RE-complete). y

In this sense Thue systems are too powerful to be useful as a complexity-theoretic lower bound, since Gröbner bases and ideal membership are computable (even in EXPSPACE). But if we move from strings (with non-commuting letters) to *commutative* strings, then the corresponding Thue systems do become decidable.

Definition 3.3 (, Commutative Thue system). Let Σ be a finite alphabet, then Σ^* is the *free commutative monoid* over

$$\langle \Sigma \mid \text{Mon}^1 \circ \cdot, \cdot - D \ E \text{ if and only if } |D|_G = |E|_G \ \forall G \in \Sigma^* \rangle$$

A *commutative Thue system* is a Thue system $\langle \Sigma^*, P \rangle$ defined over commutative strings Σ^* .

Convention: From now on only commutative Thue systems will be considered. When specifying P as a list of rules of the form

$$L \rightarrow R \quad ! \quad (\sim)$$

we implicitly always add the converse rules, too. We write $L \rightarrow R$ for a forward application of the rule and $R \rightarrow L$ for an application of the reverse rule, but we stress that \rightarrow_P is a *symmetric* relation. If the concrete derivation is not important, then we simply write \rightarrow_P . If the Thue system is understood from the context, then we omit the subscript. y

Remark. In the literature commutative Thue systems are defined as “ordinary” Thue systems containing the rules $GH \rightarrow HG$ for all $G, H \in \Sigma$, which is of course equivalent to our definition.

Commutative Thue systems can be understood as “monomial replacement systems”; this insight is a key ingredient in the reduction to the ideal membership problem. They also coincide with finite presentations of commutative semigroups in the following sense:

Example 3.4 (Commutative Thue systems “are” finite commutative monoid presentations). If P is a commutative Thue system over Σ , then $\langle \Sigma^*, \rightarrow_P \rangle$ is a monoid generated by the classes of elements of Σ^* , and commutativity of the strings ensures that $\langle \Sigma^*, \rightarrow_P \rangle$ is commutative. This commutative monoid coincides with the finitely presented monoid $\langle \Sigma^*, \equiv_P \rangle$, since the relation \equiv_P is the congruence relation generated by P (Lemma A.6).

Conversely, if $\langle \Sigma^*, \equiv_P \rangle$ is a finitely presented monoid, then by adding the reverse rules we get a commutative Thue system P such that $\langle \Sigma^*, \rightarrow_P \rangle \cong \langle \Sigma^*, \equiv_P \rangle$. y

We finally define the word problem for finitely presented commutative semigroups.

Definition 3.5 (Word problem for commutative semigroups, CSG).

- *Input:* $\langle \Sigma^*, R \rangle$, where Σ is a finite set of symbols, R a list of generating congruences $f_1 \dots f_n \equiv g_{\theta=1}^B$ and $n \geq 2$.
- *Output:* Decide whether $1 \equiv_R$ in the commutative semigroup $\langle \Sigma^*, \equiv_R \rangle$.

If the input is restricted to congruences where both sides have the same length, then the corresponding problem is denoted by CSG_h .

In other words, we want to decide whether $\langle \gamma \rangle_P$, where P is the commutative Thue system obtained from R .

A first hardness result is the following construction similar to example 3.2, appearing in the survey of Mayr [32, Theorem 17]. Let M be a deterministic single-tape Turing machine working in space $5^{1=0}$, so the following portion of the initial tape content

$$\left| \begin{array}{c} \dots \\ \{Z\} \\ 5^{1=0} \end{array} \right| G_1 G_2 \dots G_n \left| \begin{array}{c} \dots \\ \{Z\} \\ 5^{1=0} \end{array} \right|$$

of size $\# B \cdot \# G \cdot 2 \cdot 5^{1=0}$ contains all cells ever visited in the course of computation of M on G . We can index the cells by $1 \dots \#$, let

$$\gamma = B \& [B_{1-\ell} \ 1 \ 2 \ f \ -0-1g \ - \ell = 1 \dots \# \ [\ ?_g \ \ell = 1 \dots \#$$

then a configuration of M in state l on position g with symbol 1_g in tape cell ℓ can be represented as a string of length $\# + 2$ over γ as

$$l?B_1 \dots B_{\#} \ 2 \ \gamma \ - \ l \ 2 \ / \ - \ ? \ 2 \ f \ ?_1 \dots \ ?_{\#} g \ B_{\ell} = B_{1_{g-\ell}} \ \ell = 1 \dots \# \cdot \quad (3.1)$$

Intuitively, we create $\#$ copies for each symbol in the working alphabet of M so that the presence of the ℓ -th copy indicates that the corresponding symbol is currently in cell ℓ , and $?_g$ indicates that the head is on cell g . Let γ be the encoding of the initial configuration of M on input G and $\gamma = l_a ?_1 B_{-1} \dots B_{-\#}$, where l_a is the unique accepting state of M .

Lemma 3.6. *With the preceding notation there is a commutative Thue system P_{γ} over γ such that $\langle \gamma \rangle_{P_{\gamma}} = \gamma$ if and only if $G \in \text{CSG}_h$.*

Proof. We define P_{γ} to contain the following rules:

- (i) If $\gamma \ 1l \ - \ 0^0 = 1l^0 \ - \ 0^0 \ - \ L^0$, then add $l?_g B_{0-g} \ ! \ l^0 ?_g \ 1 B_{0-g}$ for $g = 2 \dots \#$.
- (ii) If $\gamma \ 1l \ - \ 0^0 = 1l^0 \ - \ 0^0 \ - \ R^0$, then add $l?_g B_{0-g} \ ! \ l^0 ?_{g-1} B_{0-g}$ for $g = 1 \dots \# - 1$.
- (iii) Add $l_a B_{0-g} \ ! \ l_a B_{-g}$ and $l_a ?_g \ ! \ l_a ?_1$ for all $g = 1 \dots \#$ and $0 \leq f \leq 1g$.

It is clear that any string $\langle \gamma \rangle_{P_{\gamma}} = F$ is of the form (3.1) since all rules preserve the structure. Furthermore, an application of the rules $M \ B$ (i) [(ii) corresponds to a (valid) transition of a configuration of M into the next one.

Assume $G \in \text{CSG}_h$, then after a finite number of steps the machine M reaches a configuration in state l_a , so $\langle \gamma \rangle_{P_{\gamma}} = l_a ?_1 B_1 \dots B_{\#}$. Using the rules from (iii) we can “clean up” the string to

derive G as desired:

$$I_a ?_q B_1 \cdots B_{\#} \rightarrow I_a ?_1 B_1 \cdots B_{\#} \rightarrow I_a ?_1 B_{-1} \cdots B_{\#} \rightarrow \dots \rightarrow I_a ?_1 B_{-1} \cdots B_{-\#} = G^*$$

Conversely, assume $G \rightarrow_{P_{n,=}} G$ and consider a repetition-free derivation

$$G = G_0 \rightarrow G_1 \rightarrow \dots \rightarrow G^*$$

As G contains I_a , there is a minimal $<_0 \leq 0$ such that $<_0$ contains I_a .

Claim: In this derivation $G_0 \rightarrow_{P_{n,=}} <_0$ only (forward) rules from \mathcal{M} are applied.

Indeed, rules from (iii) and their converse are not applicable by minimality of $<_0$. Assume that a converse rule from \mathcal{M} is used in some $G_{i+1} \rightarrow_{P_{n,=}} G_i$ and choose i maximal with this property. $G_i < <_0$, as I_a does not occur on any left hand side in \mathcal{M} (I_a is a final state), so $G_{i+1} \rightarrow_{P_{n,=}} G_i$ applies a forward rule from \mathcal{M} by maximality of i . We are in the following situation:

$$G_{i+1} \left(\mathcal{M} \text{ using } I^0 ?_q B_{0-q} \rightarrow I ?_q B_{0-q} \rightarrow G_i \right) \mathcal{M} \text{ using } I ?_q B_{0-q} \rightarrow I^0 ?_q B_{0-q}$$

For any configuration there is (at most) a single applicable rule from \mathcal{M} corresponding to the deterministic transition $q \rightarrow I - 0^q$. In this situation the rule is determined by I as both share I , $?_q$ and therefore B_{0-q} , so *the same production* is applied and the reverse and forward application cancel each other. Thus $G_{i+1} = G_i$, but the derivation was assumed to be repetition-free, so no reverse rule could have been applied in the first place, proving the claim.

We have established that $G = G_0 \rightarrow_{P_{n,=}} <_0$ consists of forward rules from \mathcal{M} corresponding to the behavior of \mathcal{M} on G . Since $<_0$ contains the accepting state, we see that $G \in !^1 \mathcal{M}^0$ and the proof is complete.

If the function f is bounded by a polynomial, then this yields a polynomial-time reduction from $!^1 \mathcal{M}^0$ to the word problem for commutative semigroups.

Theorem 3.7 (CSG_h is PSPACE-hard). *If $\mathcal{L} \in \text{PSPACE}$, then $\text{P}_{\text{m}} \text{CSG}_{\text{h}}$, in particular CSG and CSG_h are PSPACE-hard.*

Proof. Let $\mathcal{L} \in \text{PSPACE}$ be a language over $\{0, 1\}$ and let \mathcal{M} be a deterministic 1-tape Turing machine deciding \mathcal{L} in space bounded by $f(n) = 2^n$. Let $\mathcal{M}^0 = \{ \dots \}$ be the commutative presentation from the previous lemma (i.e. containing the rules (i)–(iii)) for $\# = f(n) = 2^n$. Notice that by construction, all rule are homogeneous. The map

$$f(n) \text{ B } \{ \dots \} \rightarrow \{ \dots \} \text{ B } f(n)$$

can be computed in deterministic polynomial time and reduces \mathcal{M}^0 to CSG_h by Lemma 3.6.

It turns out that CSG is *not* in PSPACE, in contrast to CSG_h , but rather EXPSPACE-hard. Our approach here does not generalize to this case, as we need a variable for each tape cell and such an alphabet cannot be written down by a polynomial-time algorithm. On the other hand, non-homogeneous rules allow the derivation strings of superpolynomial length (compared to the input string and the set of rules), see section 3.4. Thus our next step is to find a model of computation whose configurations can be encoded with long strings from Σ^* for small n .

3.2 Counter machines

It turns out that a convenient model of computation to be simulated by commutative semi-group presentations are *Counter machines*. A classical reference is Minsky's book [36, Chapter 11 and 14], where these machines are called *program machines* (and use a slightly different instruction set). Informally, these are finite automata together with a fixed number of counters. The counters can be INCREASED, DECREASED or used to BRANCH the program flow on a ZERO. More formally:

Definition 3.8 (Counter machine). A counter machine with $k \geq 1$ counters is a tuple (Q, δ, q_0, q_a) where

- Q is a finite set of states
- δ is a transition function

$$\delta : Q \times \{ \text{INC}_1, \dots, \text{INC}_k, \text{DEC}_1, \dots, \text{DEC}_k, \text{BZ}_1, \dots, \text{BZ}_k \} \rightarrow Q \times \mathbb{Z}^k$$

- $q_0 \in Q$ is the initial state and $q_a \in Q$ is the final/accepting state.

A configuration of M is described by the current state and the content of the k counters, i. e. a tuple in $Q \times \mathbb{Z}^k$. The transition function is used to define the (unique) successive configuration as follows: Let $(q, z_1, \dots, z_k) \in Q \times \mathbb{Z}^k$ be the current configuration, $q < q_a$.

- (i) If $(q, z_1, \dots, z_k) \in \text{INC-DEC}$, then the next configuration is

$$(q, z_1, \dots, z_k) \xrightarrow{\delta} (q', z'_1, \dots, z'_k) \quad \text{where } z'_\ell = \begin{cases} z_\ell + 1 & \text{if } \ell = i \text{ and op} = \text{INC} \\ z_\ell - 1 & \text{if } \ell = i \text{ and op} = \text{DEC} \\ z_\ell & \text{if } \ell < i \end{cases}$$

- (ii) If $(q, z_1, \dots, z_k) \in \text{BZ}$, the next configuration is

$$(q, z_1, \dots, z_k) \xrightarrow{\delta} \begin{cases} (q, z_1, \dots, z_k) & \text{if } z_i = 0 \\ (q', z'_1, \dots, z'_k) & \text{if } z_i < 0 \end{cases}$$

As usual, we denote the transitive and reflexive closure of the transition relation as $\overset{*}{\rightarrow}$.

Example 3.9 (Multiplication and division by 2). In this example we assume that counter 1 holds a non-negative integer and counter 2 is empty.

(i) We can move the content of counter 1 to counter 2 (emptying the former) and transition from state q to q^0 with the following instructions:

$$q \Vdash \text{BZ}_1 - q^0 - 0 \quad 0 \Vdash \text{INC}_2 - 1^0 - 1 \quad 1 \Vdash \text{DEC}_1 - q^0.$$

This has the effect of $1_{q-} \# - 0^0 \overset{*}{\rightarrow} 1_{q^0-} 0 - \#^0$.

(ii) We can similarly double the content of counter 1 if we increase twice instead of once:

$$q \Vdash \text{BZ}_1 - q^0 - 2^0 - 2 \quad 2 \Vdash \text{INC}_2 - 3^0 - 3 \quad 3 \Vdash \text{INC}_2 - 4^0 - 4 \quad 4 \Vdash \text{DEC}_1 - q^0.$$

This has the effect of $1_{q-} \# - 0^0 \overset{*}{\rightarrow} 1_{q^0-} 0 - 2\#^0$. Using the copy instructions from (i) we can then move the result back to counter 1.

(iii) Using the same trick we can also halve the first counter and branch into q_{even} or q_{odd} depending on the remainder:

$$q \Vdash \text{BZ}_1 - q_{\text{even}} - 5^0 - 5 \quad 5 \Vdash \text{DEC}_1 - 6^0 - 6 \quad 6 \Vdash \text{BZ}_1 - q_{\text{odd}} - 7^0 - 7 \quad 7 \Vdash \text{DEC}_1 - 8^0 - 8 \quad 8 \Vdash \text{INC}_2 - q^0.$$

This has the effect of $1_{q-} 2\# - 0^0 \overset{*}{\rightarrow} 1_{q_{\text{even}}-} 0 - \#^0$ and $1_{q-} 2\# \overset{*}{\rightarrow} 1 - 0^0 \overset{*}{\rightarrow} 1_{q_{\text{odd}}-} 0 - \#^0$ and we can again move the result back if desired. y

We say that the counter machine M accepts an input $\# \in 2^*$ if and only if

$$1_{q_0-} \# - 0 - \dots - 0^0 \overset{*}{\rightarrow} 1_{q_a-} 0 - \dots - 0^0.$$

Lemma 3.10 (Counter machines are Turing complete). For any Turing machine M on the input alphabet $\Sigma = \{0, 1\}$ there exists a k -counter machine M' such that M' halts on $G \in \Sigma^*$ if and only if M accepts the binary number $\# = 1G$.

A leading 1 is necessary to discriminate different strings that would otherwise describe the same binary number, for example 0 vs 00. The proof uses *stack machine* as an intermediary machine model, for our purposes, this is a finite state machine together with (in our case) two stacks Γ_1, Γ_2 . Depending on the current state and the top symbols of the stacks (including observing an empty stack $\#$), the machine transitions into a unique successor state and can *push* and *pop*¹ single elements from both stacks:

$$\delta: \Sigma \times \Gamma_1 \times \Gamma_2 \times \{ \#, \#^2 \} \rightarrow \Sigma \times \Gamma_1 \times \Gamma_2 \times \{ \text{push}^1, \text{push}^{11}, \text{pop} \}$$

¹A pop operation on an empty stack has no effect.

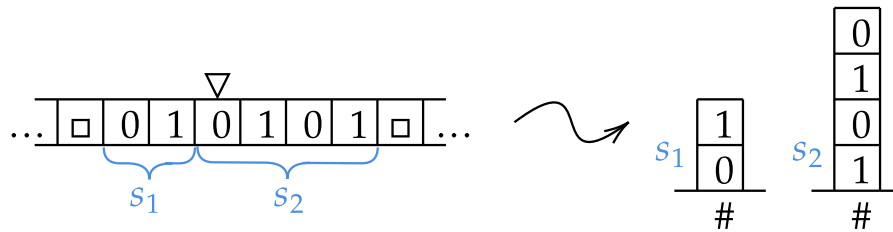


Figure 3.2: How to represent the tape of a Turing machine with two stacks.

Proof of Lemma 3.1. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \lambda, \rho)$ be the Turing machine in question, so $M : \Sigma^* \rightarrow \Gamma^* \rightarrow \Sigma^*$.

Step 1: The behavior of M can be simulated by two stacks.

We translate M and G into a stack machine as follows: The set of states is $Q \cup \{q_0\}$. If the tape content is $\dots 0_1 0_2 \dots 0_{\ell} \dots$ with head at position ℓ , then σ_1 contains the left portion $0_1 0_2 \dots 0_{\ell-1}$, σ_2 the right portion $0_{\ell+1} \dots 0_{\ell+n}$, see Figure 3.2. We translate M into \mathcal{M} as follows:

$$\begin{aligned} \mathcal{M} : Q \rightarrow Q &= \{ \sigma_1 \cdot \text{push } 0; \sigma_2 \cdot \text{pop} \} \\ \mathcal{M} : Q \rightarrow Q &= \{ \sigma_1 \cdot \text{pop}; \sigma_2 \cdot \text{pop}; \sigma_2 \cdot \text{push } 0; \sigma_2 \cdot \text{push } 1 \} \end{aligned}$$

(if $2 = \#$ then omit the last push). The starting configuration is $(\sigma_1 = \epsilon, \sigma_2 = G_1 G_2 \dots G_n \#)$.

Step 2: The behavior of a single stack can be simulated by two counters.

A stack with content $0_1 \dots 0_{\ell} \#$ is represented by a counter c_1 with value $10_1 0_2 \dots 0_{\ell}$ in binary. Pushing a $0 \cdot 1$ is performed by doubling using a second empty counter c_2 as in Example 3.9 and adding 1 if desired. Peeking at the stack corresponds to extracting the lowest bit, we implement this as in Example 3.9 together with a check if the counter has value 1 (i. e. represents the empty stack). A pop is performed by checking if the stack is nonempty ($c_1 > 1$) and then dividing by 2.

Step 3: Translating \mathcal{M} into a counter machine.

Two stacks can be simulated using three counters if c_1, c_2 hold the actual data and c_3 is used as storage for the stack operations as described in step 2. Thus it is straightforward (although a bit tedious) to translate the stack machine behavior \mathcal{M} from step 1 into a transition function for a counter machine. At the start of computation we initialize the second counter (with a single increment) and then “flip” the content of counter c_1 to c_2 in order to have the leftmost bit of the input as the least significant bit of the value of c_2 (compare step 2), then counter c_1 and c_2 precisely represent the initial configuration of the stack machine. Finally we add instructions to empty all counters if the state l_a is reached, and then transition to q_a .

Remark. In the sequel we only need this result for n -counter machines for some (fixed) $n \geq 0$ (although the construction of P is somewhat simpler if $n \geq 4$, see below), the lemma shows that 3 counters suffice. It turns out that 2-counter machines are already universal, but only if

the input is encoded in a *very* specific way [36, Theorem 14.1-1].

The computation of $\langle M \rangle$ on input $\langle x \rangle$ is said to be *bounded by* $2^{|x|}$ if the counters of *all* intermediate configurations $\langle q_0 \rangle, \langle q_1 \rangle, \dots, \langle q_n \rangle$ are in the range $0 \leq c_i \leq 2^{|x|}$. Notice that the construction in the previous lemma has the following property: If $\langle M \rangle$ is operating in space $f(n)$, then the constructed counter machine will have its counters bounded by $2^{f(n)}$, as this is a bound on the number $f(n)$, where $f(n)$ is the (binary) number spelled out on the tape.

Theorem 3.11 (EBC is EXPSPACE-complete). *The language of exponentially bounded counter machines EBC is EXPSPACE-complete:*

- *Input:* $\langle M \rangle = \langle q_0, \delta, q_a \rangle$, a k -counter-machine
- *Output:* Decide whether $\langle M \rangle$ accepts 0 and has computation bounded by $2^{2^{|x|}}$

Proof. Step 1: EBC is in EXPSPACE.

A Turing machine can easily simulate the configuration transitions. Each counter and the current state is stored as a binary number on a separate tape and the instructions INC-DEC-BE can be executed using only the cells belonging to the counters. If the computation of $\langle M \rangle$ is bounded by $2^{2^{|x|}}$, then the space resources of this Turing machine are bounded by $3 \cdot 2^{2^{|x|}}$, which is exponential in the input length of $\langle M \rangle$.

Step 2: $\text{EBC} \stackrel{P}{m}$ EBC hard for all EXPSPACE .

Let $\langle M \rangle$ be a 1-tape machine deciding EBC which visiting at most $2^{2^{|G|}}$ cells on the working tape on input G for some fixed $k \geq 7$. In order to show $\text{EBC} \stackrel{P}{m}$ EBC we use the 3-counter machine constructed in the previous lemma. For $G \in \{0, 1\}^*$ let $\langle M_G \rangle$ be the following 3-counter machine:

- (i) Starting from q_0 use δ^k instructions to count 2_1 to the value $1G$ using the binary representation and the doubling algorithm from example 3.9.
- (ii) Then add the instructions from Lemma 3.10 to simulate the behavior of $\langle M \rangle$.
- (iii) Artificially enlarge the set of states Q (if necessary) in order to ensure $2^{2^{|G|}} \leq |Q| \leq 2^{2^{|G|} + 1}$.

Then the computation of $\langle M_G \rangle$ is bounded by $2^{2^{2^{|G|} + 1}} = 2^{2^{2^{|G|} + 1}}$ and $\langle M_G \rangle$ accepts 0 if and only if $G \in \text{EBC}$. Step (ii) only depends on $\langle M \rangle$ and not on G , so the map $G \mapsto \langle M_G \rangle$ can be computed in polynomial time.

Remark. It may seem kind of arbitrary to consider the computation bounded by $2^{2^{|x|}}$. Indeed, if we require $\langle M \rangle$ to have its computation bounded by $2^{|x|}$ instead, then this problem becomes PSPACE-complete.

- (i) Σ contains $(\sigma, \delta, \delta = 1 \dots 4)$ among other symbols.
- (ii) The only words containing (or derivable from) $(\sigma F, F \neq \sigma)$ are (σF) and $(\sigma \sigma^{2^2} F)$.
- (iii) Similarly, the only words containing (or derivable from) (σF) are (σF) and (σF^0) with $F = F^0 \sigma^{2^2}$.

With this we can “compress” the presentation P^0 to P where Let

$$B = \{ \sigma \} \cup \{ \sigma^i \mid i \in \mathbb{N} \} \cup \{ \sigma^i \sigma^j \mid i, j \in \mathbb{N} \} \cup \{ \sigma^i \sigma^j \sigma^k \mid i, j, k \in \mathbb{N} \} \cup \{ \sigma^i \sigma^j \sigma^k \sigma^l \mid i, j, k, l \in \mathbb{N} \}.$$

Let $e = B^{2^2}$, then Σ contains the following symbols with specific purpose:

- Σ contains σ to create the strings of double-exponential length.
 $\{ P \text{ contains } P_{\sigma}$.
- Σ contains the symbols from Σ^0 encoding the configurations of Σ .
 $\{ P \text{ contains } P^0 \text{ except for the (BE) rules } \sigma^i \sigma^j \rightarrow \sigma^i \sigma^j \sigma^k$.
- For each branch-on-zero state $\sigma \in \Sigma$ & Σ contains two auxiliary “states” whose purpose is to verify that the configuration contains σ^{2^2} by breaking down this string ($\sigma^{\#}$) and then building it back up and transitioning to the successor state (σ^{\cdot})
 $\{ P \text{ contains for each } \sigma \in \Sigma \text{ & with } \sigma^i \sigma^j \rightarrow \sigma^i \sigma^j \sigma^k \text{ the rules}$

$$\sigma^i \sigma^j \rightarrow \sigma^i \sigma^{\#} \sigma^j \quad (\&)$$

$$\sigma^{\#} \sigma^i \rightarrow \sigma^{\cdot} \sigma^i \quad (\times)$$

$$\sigma^{\cdot} \sigma^i \rightarrow \sigma^i \sigma^{\cdot} \quad (\%)$$

- Finally, Σ contains $\sigma_0 \dots \sigma_{a-3}$ to expand the initial and collaps the final configuration.
 $\{ P \text{ contains the rules}$

$$\sigma_0 \rightarrow \sigma_0 \sigma_1 \quad (0_0)$$

$$\sigma_0 \sigma_1 \rightarrow \sigma_0 \sigma_2 \quad (0_1)$$

$$\sigma_0 \sigma_2 \rightarrow \sigma_0 \sigma_3 \quad (0_2)$$

$$\sigma_0 \sigma_3 \rightarrow \sigma_0 \quad (0_3)$$

$$\sigma_a \rightarrow \sigma_{a-3} \sigma_3 \quad (a_3)$$

$$\sigma_{a-3} \sigma_3 \rightarrow \sigma_{a-2} \sigma_2 \quad (a_2)$$

$$\sigma_{a-2} \sigma_2 \rightarrow \sigma_{a-1} \sigma_1 \quad (a_1)$$

$$\sigma_{a-1} \sigma_1 \rightarrow \sigma_{a-0} \sigma_0 \quad (a_0)$$

We first note that this new presentation P still behaves as P^0 .

Lemma 3.14. *We have the following equivalences:*

$$(i) \ @_{0-0} \ P \ @_0 \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} e_- \\ 2 \end{matrix} \ \begin{matrix} e_- \\ 3 \end{matrix} = \text{rep}^1_{@_0-0-0-0^0} \text{ and } @_{a-0} \ P \ @_a \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} e_- \\ 2 \end{matrix} \ \begin{matrix} e_- \\ 3 \end{matrix} = \text{rep}^1_{@_a-0-0-0^0}.$$

$$(ii) \ \text{If } \text{rep}^1_{@-2_1-2_2-2_3^0} \ P^0 \ \text{rep}^1_{@^\ell-2_1^0-2_2^0-2_3^0}, \text{ then } \text{rep}^1_{@-2_1-2_2-2_3^0} \ P \ \text{rep}^1_{@^\ell-2_1^0-2_2^0-2_3^0}.$$

Proof. (i) This is a consequence of (0₀)–(0₃) and Theorem 3.13:

$$\begin{aligned} @_{0-0} \ \begin{matrix} (0_0) \\ \end{matrix} \ @_{0-1} \ \begin{matrix} (3.13) \\ \end{matrix} \ @_{0-1} \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} (0_1) \\ \end{matrix} \ @_{0-2} \ \begin{matrix} (3.13) \\ \end{matrix} \ @_{0-2} \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} e_- \\ 2 \end{matrix} \\ \begin{matrix} (0_2) \\ \end{matrix} \ @_{0-3} \ \begin{matrix} (3.13) \\ \end{matrix} \ @_{0-3} \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} e_- \\ 2 \end{matrix} \ \begin{matrix} e_- \\ 3 \end{matrix} \ \begin{matrix} (0_3) \\ \end{matrix} \ @_0 \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} e_- \\ 2 \end{matrix} \ \begin{matrix} e_- \\ 3 \end{matrix} \end{aligned}$$

The second derivation is completely analogous with (a₀)–(a₃) instead.

(ii) P inherits all rules from P except for $@ \ \begin{matrix} e_- \\ \ell \end{matrix} \ ! \ @^\ell \ \begin{matrix} e_- \\ \ell \end{matrix}$. If this rule is applied in a derivation (say with $\ell = 1$) to $\text{rep}^1_{@-e_- -2_2-2_3^0} \ P^0 \ \text{rep}^1_{@^\ell-e_- -2_2-2_3^0}$, then we have

$$\begin{aligned} \text{rep}^1_{@-e_- -2_2-2_3^0} = @ \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} (\&) \\ \end{matrix} \ @_\# \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} (3.13) \\ \end{matrix} \ @_\# \ \begin{matrix} \end{matrix} \ \begin{matrix} (\times) \\ \end{matrix} \ @_\cdot \ \begin{matrix} \end{matrix} \\ \begin{matrix} (3.13) \\ \end{matrix} \ @_\cdot \ \begin{matrix} e_- \\ 1 \end{matrix} \ \begin{matrix} (\%) \\ \end{matrix} \ @^\ell \ \begin{matrix} e_- \\ 1 \end{matrix} = \text{rep}^1_{@^\ell-e_- -2_2-2_3^0}. \end{aligned}$$

The difficult part is to prove that the inverse is also true: An equivalence of configurations by P implies the equivalence by P^0 . Consider the set of all possible configurations

$$, = \text{rep}^1_{@-2_1-2_2-2_3^0} \ @ \ 2 \ \&- \ 0 \ 2_1-2_2-2_3 \ e_- \cdot$$

Lemma 3.15. (i) *Any derivation $@_{0-0} \ P \ @_{a-0}$ contains $\text{rep}^1_{@_0-0-0-0^0}$ and $\text{rep}^1_{@_a-0-0-0^0}$.*

(ii) *For all $F \ F^0 \ 2$, we have $F \ P \ F^0$ if and only if $F \ P^0 \ F$.*

With this result we can finally prove the promised hardness result for CSG.

Theorem 3.16 (CSG is EXPSPACE-hard). *The map $\mathbb{T} \ 1 \ -P \ -@_{0-0} \ -@_{a-0}^0$ defines a polynomial-time many-one reduction $\text{EBC} \ \xrightarrow{P} \ \text{CSG}$. In particular, CSG is EXPSPACE-hard.*

Proof. The map can clearly be computed in polynomial time (provided this is true for P_\cdot , which we will see in the next section). If $\ 2 \ \text{EBC}$, then $\text{rep}^1_{@_0-0-0-0^0} \ P^0 \ \text{rep}^1_{@_a-0-0-0^0}$ by Lemma 3.12 and consequently $@_{0-0} \ P \ @_{a-0}$ by Lemma 3.14. Conversely, if $@_{0-0} \ P \ @_{a-0}$, then by Lemma 3.15 and again Lemma 3.12 $\ 2 \ \text{EBC}$.

Proof of Lemma . . . (i) Any derivation $@_{0-0} \ P \ @_{a-0}$ must replace $@_{0-0}$, the only rule allowing for this is (0₀), introducing $@_{0-1} \ (\ 1$. Only the rules (0₀)–(0₃) and P_\cdot are applicable until the first word containing some $@ \ 2 \ \&$ is produced (which will be $@_0$ by (0₃)).

In order to apply (0₁) the previous rules from P₌ must create a string containing ϵ , by 3.13 the only possibility being $\epsilon_1 \epsilon_1$. If the derivation is repetition-free, then the only applicable rule now is (0₁), so the derivation has the form

$$\epsilon_{0-0} \xrightarrow{(0_0)} \epsilon_{0-1} (\epsilon_1) \xrightarrow{P=} \epsilon_{0-1} \epsilon_1 \xrightarrow{(0_1)} \epsilon_{0-2} (\epsilon_2 \epsilon_1).$$

Repeating this argument two more times shows that the word $\epsilon_0 \epsilon_1 \epsilon_2 \epsilon_3$ must occur in the derivation $\epsilon_{0-0} \xrightarrow{P} \epsilon_{a-0}$. Essentially the same reasoning starting from ϵ_{a-0} and using (a₀)–(a₃) shows that $\epsilon_a \epsilon_1 \epsilon_2 \epsilon_3$ is contained in the derivation, too.

(ii) The backward implication was proven in Lemma 3.15, so assume $\epsilon = \text{rep}^1 \epsilon_{-2_1-2_2-2_3^0} \xrightarrow{P} \text{rep}^1 \epsilon_{-2_1^0-2_2^0-2_3^0} = \epsilon$. We proceed by induction on the length of a derivation, the case of length 0 being trivial. Consider a shortest (necessarily repetition-free) derivation

$$\epsilon = \epsilon_0 \xrightarrow{P} \epsilon_1 \xrightarrow{P} \dots \xrightarrow{P} \epsilon_n = \epsilon$$

and let $\epsilon_{<} \xrightarrow{P_A} \epsilon_{<+1}$ be the first step using one of the “new” rules (&), ..., (a₀). Then all previous steps in the derivation $\epsilon_0 \dots \epsilon_{<}$ consists of rules from P⁰ (the rules from P₌ cannot be applied in the absence of ϵ), so $\epsilon_{<} : 2$, and $\text{rep}^1 \epsilon_{-2_1-2_2-2_3^0} \xrightarrow{P^0} \epsilon_{<}$.

The only rules possibly applicable to $\epsilon_{<}$ are (&),(%),(0₃),(a₃), as all other rules require the presence of symbols not in $\epsilon_{<} : 2$.

(0₃) Then $\epsilon_{<}$ contains state @₀ which is replaced by @₀₋₃, and hence the only rules applicable to $\epsilon_{<+1}$ containing @_{0-3} \epsilon_3 are rules from P₌. Theorem 3.13 tells us that P₌ can only be used to produce @_{0-3} (\epsilon_3, furthermore, the maximum number e₌ of \epsilon_3's are removed in this process (and so the string contains neither \epsilon_3's nor \epsilon_3's). Then the only applicable rule is (0₂) and we can repeat this argument twice to arrive at @_{0-1} (\epsilon_1. Now the only applicable rule replaces this with @_{0-0} at which point no rule can be applied.}}}}

(a₃) The same line of reasoning prohibits the occurrence of this rule in a repetition-free derivation.

(&) Again only rules from P₌ can be applied to $\epsilon_{<+1}$ and by applying Theorem 3.13 twice we must have

$$\begin{aligned} \epsilon_{<} = @ \epsilon_{\delta}^{e_{=}} F \xrightarrow{(\&)} \epsilon_{<+1} = @_{\#} \epsilon_{\delta}^{e_{=}} F \xrightarrow{P=} @_{\#} (\epsilon_{\delta} F \\ \xrightarrow{(\times)} @_{\#} (\epsilon_{\delta} F) \xrightarrow{P=} @_{\#} \epsilon_{\delta}^{e_{=}} F \xrightarrow{(\%)} @_{\delta}^{e_{=}} F = \epsilon_{<+1}; \end{aligned}$$

where @⁰ is determined by $\epsilon_{<}^1 @^0$ and F determined by $\epsilon_{<}$. Hence $\epsilon_{<} \xrightarrow{P^0} \epsilon_{<+1} : 2$.

(%) The same argument yields $\epsilon_{<} \xrightarrow{P^0} \epsilon_{<+1} : 2$, for some $\epsilon_{<} : 1$.

By induction $\epsilon_{<+1} : P^0$ and thus $\epsilon_{<} : P^0$.

3.4 Producing words of double-exponential length

In this section we prove Theorem 3.13, we follow the exposition by Bayer & Stillman [4]. Let $e = B^{2^A}$. The alphabet Σ consists of $10^{1-A} = 10^{A-1}$ characters

$$\Sigma = \bigcup_{A=0}^{\infty} \Sigma_A \quad \Sigma_A = \{B_{A-5}, 1_{A-1}, 1_{A-2}, 1_{A-3}, 1_{A-4}, 2_{A-1}, 2_{A-2}, 2_{A-3}, 2_{A-4}\}.$$

Definition 3.17 (Level of a word, box). The symbols in Σ_A are defined to be of *level* A . A word $F \in \Sigma^*$ is said to be of *level* A if all symbols in F are of level $\leq A$ and

$$|F|_{\Sigma_{A-5}} = |F|_{\Sigma_A} = 1, \quad |F|_{\Sigma_{2A-8}} = 1, \quad |F|_{\Sigma_{\theta}} = |F|_{\Sigma_{5\theta}} = 0 \quad \text{for } \theta \geq A.$$

The set of level A words is $\text{LVL}^1 A^0 / \text{LVL}^1 A^0$, if $F \in \text{LVL}^1 A^0$, then denote by $\text{box}^1 G^0$ the (unique) combination of $B \cdot 5$ and 2_{θ} in G .

Notice that not all words are assigned a level. In order to ease notation, when fixing A we use $(- \cdot -)_{\theta}$ to denote the symbols of level A and $B \cdot 5 - 1_{\theta} - 2_{\theta}$ for level $A - 1$ respectively. The defining relations in P_{Σ} are defined inductively as follows: P_0 contains the four rules

$$B_{0-2_{\theta}} \rightarrow 5_{0-2_{\theta}} 1_{0-\theta}^2, \quad \theta = 1, \dots, 4. \quad (O_{\theta})$$

For $A \geq 1$ define P_A to contain P_{A-1} and the ten rules

$$(A) \quad B_{A-1} \rightarrow B_{2A-1} \quad (A)$$

$$(B) \quad 5_{2A-1} \rightarrow B_{2A-2} \quad (B)$$

$$(C_{\theta}) \quad 5_{2A-2_{\theta}} 1_{2A-2_{\theta}} \rightarrow 5_{2A-2_{\theta}} 1_{2A-2_{\theta}} \quad \theta = 1, \dots, 4 \quad (C_{\theta})$$

$$(D) \quad 5_{2A-1_1} \rightarrow 5_{2A-1_4} \quad (D)$$

$$(E) \quad B_{2A-3} \rightarrow B_{2A-2} \quad (E)$$

$$(F) \quad B_{2A-3} \rightarrow 5_{2A-4} \quad (F)$$

$$(G) \quad B_{2A-4} \rightarrow \quad (G)$$

The rules move words (of a given level) around in the boxes, this can be visualized as in figure 3.3. We first verify that the string e_{θ} (from level $A = \infty$) can actually be derived from $(- \cdot -)_{\theta}$ in the commutative Thue system defined by P_{Σ} over Σ .

Lemma 3.18. For $A = 0, \dots, \infty$ and $\theta = 1, \dots, 4$ we have

$$B_{A-2_{\theta-A}} \xrightarrow{P_{\Sigma}} 5_{A-2_{\theta-A}} 1_{\theta-A}^{e_A}. \quad (V)$$

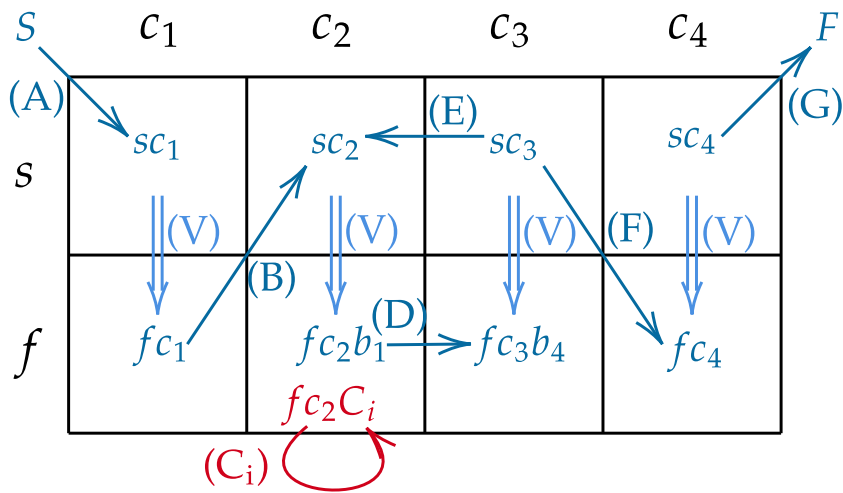


Figure 3.3: Rules (A)–(G),(V) in the context of boxes.

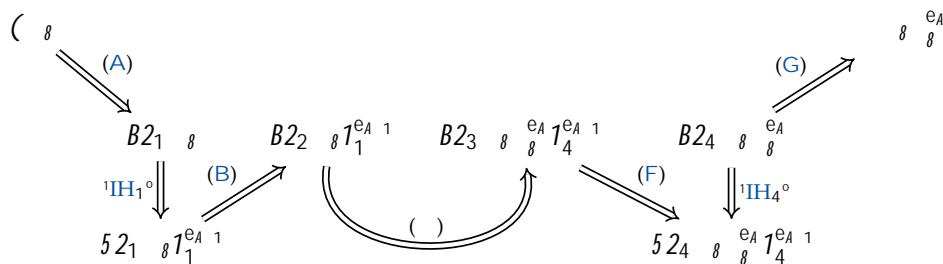
Proof. We proceed by induction on A , the case $A = 0$ (and $e_0 = 2$) follows by applying (O_δ) . Now considering level $A - 1$, we assume the induction hypothesis $(IH_\delta) B_{2\delta} \stackrel{P}{=} 52_\delta 1_\delta^{e_A - 1}$ for $\delta = 1 \dots 4$. We can “exchange” 1_1 for ${}^{e_A - 1}1_4$ in the presence of $B_{2\delta}$ as follows:

$$\begin{aligned}
 B_{2\delta} 1_1 &\stackrel{(A)}{=} 52_\delta 1_1 1_2^{e_A - 1} \stackrel{(B)}{=} 52_\delta 1_1 1_3^{e_A - 1} 1_4^{e_A - 1} \\
 &\stackrel{(C)}{=} 52_\delta 1_1 1_3^{e_A - 1} 1_4^{e_A - 1} \stackrel{(D)}{=} 52_\delta 1_1 1_3^{e_A - 1} 1_4^{e_A - 1} \stackrel{(E)}{=} B_{2\delta} 1_1 1_4^{e_A - 1}
 \end{aligned}$$

If we repeat this sequence of derivations $e_A - 1$ -many times except for the final application of (E) , then we obtain

$$B_{2\delta} 1_1^{e_A - 1} \stackrel{P}{=} B_{2\delta} 1_1^{e_A - 1} 1_4^{e_A - 1} \quad ()$$

With this in mind we can finally prove the assertion, following figure 3.3



Remark. The derivation $()_P \stackrel{P}{=} {}^{e_A}$ in the proof is extremely long, this is not a surprise, as we need to derive a string of length 2^{2^e} and each individual rule only increases the length of the string by at most 2 symbols. To be precise, let $=$ denote the length of above derivation,

	2_1	2_2	2_3	2_4
B	$10-0^0$	$14-0^0$	$14-0^0$	$10-0^0$
5	$14-0^0$	$14-4^0$	$14-4^0$	$14-0^0$

Table 3.1: Values of β on the boxes, $4 \leq B \leq 1$.

Lemma 3.21. *Let $F \in \text{LVL}^1 A^0$ and assume Theorem 3.20(i)+(iii) is true in the case $A \leq 1$. If there is a single $G_0 \in \text{LVL}^1 A \leq 1 \setminus \gg F|_{P_A}$ such that the value $\beta_{G_0^0}$ is according to box $\beta_{G_0^0}$ in table 3.1, then this is true for all $G \in \text{LVL}^1 A \leq 1 \setminus \gg F|_{P_A}$. The same is true for $G^1 Q_A^0$.*

Proof. Consider a simple path in $G^1 P_A^0$ from G_0 to $G \in \text{LVL}^1 A \leq 1$. This path is an alternating sequence of segments of rules $P_A \cap P_{A-1}$ and of segments contained in $G^1 P_{A-1}^0$.

- Rules (A) or (G) do not appear on the path, as this would result in a word without $2_{\theta, \delta}$ for $\theta = A-1$, and no rule of P_A can be applied (except for (A) or (G) again but the path is simple).
- Rules (B)–(F) preserve being of level $A-1$ and also preserve correctness of table 3.1.
- Segments in $G^1 P_{A-1}^0$ preserve being of level $A-1$ by Theorem 3.20(i) and must end in a word of level $A-1$. Theorem 3.20(iii) then shows that the start and end of this segment also respect the values in Table 3.1.

Thus, since the statement is true for G_0 and is preserved along the path, it holds true for G . The proof for $G^1 Q_A^0$ is essentially the same, as rule (C) also respects the table.

Proof of Theorem 3.20. We proceed by induction on A , starting with $A = 0$. The arguments for $G^1 Q_A^0$ only depend on the induction hypothesis for $G^1 P_A^0$, hence the start only needs to consider P_0 .

If F contains $2_{0, \delta}$ and B_0 or 5_0 , then there is at most one applicable rule, namely (O_δ) in one direction. This rule clearly preserves the property of being of level 0. Since there is at most one applicable rule, we immediately get that the component of F is either a single word or a pair of words separated by one application of (O_δ) , this proves both (ii) and (iii).

Now let $A \leq 1$ and assume that the Theorem is proven for the case $A \leq 1$.

(i) Depending on whether F contains $($ or $)$, the rule (A) or (G) is the only applicable rule and yields a neighbor G_0 of level $A-1$. This word has $\beta_{G_0^0} \in \{B_{2_1}, 5_{2_4}\}$ and no 1_δ 's, hence Lemma 3.21 holds true.

Let G be any word of level $A-1$ in the component of F . Any rule from P_{A-1} applied to G leads to words of level $A-1$ by induction hypothesis (i). Any rule from $P_{A-1} \cap P_A$ or (C) either leads to

- a word F^0 of level $A-1$ (in the case of (B)–(F), (C)) or

- a word F^0 containing $($ or $)$ and no symbols of level $A - 1$ except possibly $1_1 \cdots 1_4$ (in the case of (A) or (G)). In this case we must have $\text{box}^1 G^0 \in \{B_{2_1}, B_{5_2}, B_{4_3}\}$ and Lemma 3.21 tells us that G contains no 1_δ , so F^0 is of level A . Furthermore, in this case G is the only neighbor of F^0 .

Thus the words on any simple path in $G^1 P_A^0$ or $G^1 Q_A^0$ from F

$$F \rightarrow G_0 \rightarrow G \rightarrow F^0$$

are all of level $A - 1$ except possibly the last one being of level A (G is the only neighbor of F^0 , hence such a word cannot appear inside simple path). This proves (i) for $G^1 P_A^0$, in $G^1 Q_A^0$ we start at $F = ($, then F^0 must be a single $($ or $)$ as no rule of Q_A introduces other symbols of level A .

- (ii) Consider a simple cycle C in a component of $($ in $G^1 Q_A^0$, i. e.

$$G_0 \rightarrow P_A G_1 \rightarrow P_A \dots \rightarrow P_A G_{\ell-1} = G_0 \quad \ell \geq 2 \quad G_\delta < G_\eta \text{ for } 0 \leq \delta < \eta < \ell - 1.$$

Step 1: Translate into a statement purely about words of level $A - 1$.

The cycle C does not contain $($ or $)$ (having only a single neighbor), hence by (i) all G_δ are in LVL^{A-1} . At least one of the G_δ must be of level $A - 1$, since otherwise no edges stemming from $Q_A \cap P_{A-1}$ are used and C is a cycle on a component of a level $A - 1$ word inside $G^1 P_{A-1}^0$, contradicting the induction hypothesis (ii) for $G^1 P_{A-1}^0$.

Let $H_0, H_1, \dots, H_{\ell-1} = H_0 \in \text{LVL}^{A-1}$ be all words in C of level $A - 1$ (in order of appearance). As in the proof of Lemma 3.21 we see that H_η is joined to $H_{\eta-1}$ by either a rule (B),(C),(D)-(F) from $Q_A \cap P_{A-1}$ or $B_{2_\delta} \rightarrow P_{A-1} B_{5_2} 1_\delta^{e_A}$ (V) by hypothesis (iii).

Step 2: Put the H_δ into boxes and observe the transitions.

Hence no H_η is in box B_{2_1} or B_{2_4} , as words in these boxes have only one outgoing edge (case (V)). By extension, no H_η is in box B_{5_2} or B_{5_4} , since otherwise one of $H_{\eta-1}$ would necessarily be in box B_{2_1} or B_{2_4} . Hence all H_η lie in boxes $\{B_{2_2}, B_{5_2}, B_{5_3}, B_{2_3}\}$. We must have a sort of circular traversal of the boxes for the following reason:

- If $\text{box}^1 H_\eta \in \{B_{2_2}, B_{2_3}, B_{5_3}\}$, then by inspection of (B)-(F),(V)

$$\text{box}^1 H_\eta \rightarrow \text{box}^1 H_{\eta-1} = \begin{cases} \{B_{2_3}, B_{5_2}\} & \text{if } \text{box}^1 H_\eta \in \{B_{2_2}, B_{5_3}\} \\ \{B_{2_2}, B_{5_3}\} & \text{if } \text{box}^1 H_\eta = B_{2_3} \end{cases}$$

- If $\text{box}^1 H_\eta = B_{5_2}$, then by Lemma 3.21 $j H_\eta j_{1_2} \rightarrow j H_\eta j_{1_3} = e_{A-1}$. The only applicable rule within the box is (C), trading 1_2 for 1_3 . The other rules are (D) (to box B_{5_3}) and case (V) (to box B_{2_3} , only applicable if $j H_\eta j_{1_2} = e_{A-1}$). If (D) leads to, say, $H_{\eta-1}$, then the vertical rule (V) is

only applicable if $jH_{\delta-1}j_{13} = e_{A-1}$, hence H_{δ} must be part of a sequence

$$H_{\delta} \overset{(V)}{)} H_{\delta-1} \overset{(C_{\delta})}{)} \dots \overset{(C_{\delta})}{)} H_{\delta} \overset{(C_{\delta})}{)} \dots \overset{(C_{\delta})}{)} H_{A-2} \overset{(D)}{)} H_{A-1} \overset{(V)}{)} H_A.$$

$\{Z\}$
 e_{A-1} applications of (C_{δ})

Step 3: Derive a contradiction from these observations.

These considerations show that the sequence of H_{δ} 's must loop around the four boxes in a fixed direction, say $B_{22} \{ 5_{22} \{ 5_{23} \{ B_{23} \{ B_{22}$. Along such a loop, rule (D) is the only rule affecting 1_A and it strictly increases the number $jH_{\delta}j_{1A}$. But this is clearly impossible in a closed cycle, hence the component of $($ in $G^1Q_A^0$ contains no cycle.

Assume there is a simple cycle C in $G^1P_A^0$, then the homomorphism $?_A$ maps this to a cycle $?_A^1C^0$ in $G^1Q_A^0$. $?_A^1C^0$ is in the component of $?_A^1F^0$ $2 f(- g$, but this is the same component (Lemma 3.19). As any rule in P_A involves symbols not deleted by $?_A$, $?_A^1C^0$ is a non-trivial cycle in $G^1Q_A^0$, a contradiction.

(iii) The existence of a path $()_{Q_A}$ is Lemma 3.19, uniqueness is immediate from (ii), as distinct simple paths can be patched together to form a non-trivial cycle.

Let $G, H \in LVL^1 A^0$ be distinct with $G \overset{()}{P_A} H$, the uniqueness of such a path is again a consequence of (ii). Since there is some rule from P_A applicable to G , we must have $G \in LVL^1 A^0$, the same is true for H . We must have $?_A^1G^0 < ?_A^1H^0$, since otherwise the path from G to H is mapped to a nontrivial cycle in $G^1Q_A^0$, contradicting (ii). Without loss of generality, let $?_A^1G^0 = (, ?_A^1H^0 = .$

Let $\text{box}^1G^0 = (\delta$, then $\text{box}^1H^0 = \delta$, as all rules of P_A preserve the level A symbol δ . The image of the path in $G^1Q_A^0$ is a path from $($ to $.$, hence the unique sequence of derivations from Lemma 3.19. This information together with the presence of δ (and no other δ) allows us to conclude that the path from G to H uses the sequence of derivations from Lemma 3.18. In particular, if $G = (\delta$, then $H = \delta \overset{e=}{\delta}$.

Theorem 3.13 is now an easy consequence:

Proof of theorem . . . The presentation $\langle \dots \rangle_{P=}$ is of length $\$^{1=0}$ with $10^{1=}$ variables and $10^{=}$ relations of length 9 (the longest being (C_{δ})).

If $(\delta)_{P=}$ F such that F contains $.$, then F is of level $=$ and Theorem 3.20(iii) says that

$$(\delta = (\delta - F = \delta \overset{e=}{\delta} = \delta \overset{e=}{\delta}.$$

The proof of the second claim in Theorem 3.13 is analogous.

Remark. While we mostly care about $P=$ and the derivation $(\delta)_{P_A} \delta \overset{e=}{\delta}$, the structure of $G^1Q_A^0$ is used crucially in the final step of the argument. This is one explanation to why we

Step 1. $\langle = ' .$ Consider a linear combination

$$s = \sum_{i=1}^{\infty} b_i \cdot 2^i \cdot \dots \tag{3.3}$$

Let $3 \cdot 2^{\infty} \cdot 70$ be a common denominator of all coefficients of all b_i , then $3b_i \cdot 2^i \cdot \dots$. By artificially increasing the number of summands in (3.3), we may assume that all $3b_i$ are monomials with coefficient 1, for example

$$3b_i = 3 \cdot \dots \cdot 2^i \cdot \dots \{ \dots \cdot 2^i \cdot \dots \cdot 1 \cdot \dots \}$$

Thus we obtain a (potentially much larger) sum

$$3 \cdot \dots = \sum_{i=1}^{\infty} 1 \cdot 1^0 \cdot \dots \cdot 1 \cdot \dots \cdot 1 \cdot \dots \tag{3.4}$$

We claim that there is a derivation $\langle \rangle_P$ of length at most $\#$. Indeed, since the term \dots occurs on the left side of (3.4), this monomial must occur in one of the summands, i. e. $\dots = 1 \cdot \dots$. Removing this summand yields

$$3 \cdot \dots = \sum_{i=1}^{\infty} 1 \cdot 1^0 \cdot \dots \cdot 1 \cdot \dots \tag{3.5}$$

such that $\langle \rangle_P$ by application of the rule \dots . If $1 = \dots$ then we are done, otherwise apply the same reasoning to the monomial \dots to inductively obtain \dots . This process ends after at most $\#_0$ steps with $\#_0 = \dots$, since each step removes a summand from (3.4).

Step 2. $\langle = \dagger ?$. Here nearly the same proof applies: Take a linear combination as in (3.3), this time over $\dagger ?$, then we can interpret the b_i as some polynomials with integer coefficients (being unique modulo $?$). We can again rewrite this sum as in (3.4), this time with $3 = 1$ (no need to kill denominators). Then the same replacement scheme yields a sequence $\langle \rangle_P$. The left hand side always reads \dots at each step, so no cancellation can happen (even mod $?$) unless $g = \dots$ after at most $\#$ steps and we obtain a derivation $\langle \rangle_P$.

Step 3. \langle arbitrary. Any field contains either $'$ or some $\dagger ?$, its so-called *prime field*. All polynomials considered are defined over the prime field, so Lemma A.2 shows that $5 \cdot 2 \cdot \dots$ already implies $'$ or $\dagger ?$ respectively, so by the previous two cases $\langle \rangle_P$.

This shows that the equivalence relation $\langle \rangle_P$ is basically the relation \dots from Lemma 1.34 on monomials. In this way we get a reduction from commutative semigroups to polynomial ideals.

Theorem 3.23 (CSG reduces to IM_ζ). *We have $CSG \stackrel{P}{m} IM_\zeta$ for any field ζ (and even for $\zeta = \mathbb{S}$). This is also true for the homogeneous variants, i. e. $CSG_h \stackrel{P}{m} IM_{h-\zeta}$.*

Proof. Indeed, the previous theorem shows

$$1 - P - \dots \stackrel{\circ}{=} 2 \text{ CSG} \quad \text{if and only if} \quad 1 \cdot 5 - 5_1 - \dots - 5_B \stackrel{\circ}{=} 2 \text{ IM}_\zeta \cdot$$

Moreover the reduction mapping is computationally trivial, regardless of the encoding of coefficients (the only coefficients required are two constants ± 1) or monomials (the total degree is bounded).

For the second statement it suffices to notice that the $5_1 - \dots - 5_B$ are homogeneous if P consists of homogeneous congruence rules.

Together with the previous hardness results and the upper bounds from section 2.6 we see:

Corollary 3.24. *The problem IM_ζ is EXPSPACE-hard and CSG, IM_{\cdot} are EXPSPACE-complete. The homogeneous variant $IM_{h-\zeta}$ is PSPACE-hard and $CSG_h, IM_{h-\cdot}$ are PSPACE-complete.*

It is interesting to consider the analogous problem in the context of algebraic complexity theory. Bürgisser has adapted these techniques to prove analogous results for algebraic circuits.

Theorem 3.25 (Bürgisser 1998 [9]). *For any infinite field ζ there is a constant 2 such that any sequence of algebraic circuits $\{C_n\}_{n \in \mathbb{Z}}$ deciding IM_ζ has $\text{depth}(C_n) \leq 2^n$ for $n \geq 0$.*

Remark. Recall the Hermann bound which states that if $5 = \prod_{\beta=1}^B \beta 5_\beta$, then the β may be chosen of degree $\deg(5) \leq B \max_\beta \deg 5_\beta^{O(1)}$. The results from the previous section show that this bound is asymptotically tight as follows: We have seen that there is a unique derivation (\dots) which contains words of length $\leq e_n - 1$. This corresponds to a polynomial equation $(\dots) = \prod_{\beta} \beta 5_\beta$ and the proof of 3.22 shows that the degree of any term $\beta 5_\beta$ gives an upper bound on the length of words in the derivation. Thus, any such β must have degree double-exponential in n , for details see the paper by Bayer & Stillman [4, Theorem 2.4].

3.6 Church-Rosser systems

We return to the topic of Gröbner bases. The previous section gave a connection between commutative semigroup presentations and binomial ideals. We now extend this connection and define a notion analogous to Gröbner bases, following the paper by Huynh [21]. Fix a monomial order \prec on Mon_\pm and consider a commutative Thue system P .

Definition 3.26 ((Ir)reducible words). Let $P \subseteq \text{B} \langle \pm 1 \rangle; -A^0 \cup P \cup j; A \cup g$. A word $G \in \text{Mon}_\pm$ is *reducible* with respect to P if $G \succ_P H$ for some $H \in \text{Mon}_\pm$, and *irreducible* otherwise. The transformation $G \rightarrow_P H$ is called a reduction step. y

Notice that if $G \rightarrow_P H$, then by the defining property of a monomial order, $G \leq H$. As \leq is a well-order, any sequence of reduction steps must be of finite length and hence every word is equivalent to an irreducible word with respect to P . Of course, such an irreducible word is generally not unique, similarly to how normal forms of polynomials are not unique. But we have seen that this does not happen for Gröbner bases (Theorem 1.18(b)), so we use this to define an analogous notion for commutative Thue systems.

Definition 3.27 (Church-Rosser system, reduced). A commutative Thue-system $\langle \Sigma, P \rangle$ is a Church-Rosser system if for any two words $D, E \in \Sigma^*$, both irreducible with respect to P , we have $D = E$.

A Church-Rosser system is *reduced* if for all $u, v \in \Sigma^*$ both u and v are irreducible with respect to P and $u \sim_P v$ implies $u = v$.

We use the notation from the previous section to translate commutative Thue systems into ideal generators.

Theorem 3.28. Let P be a commutative Thue system and

$$B = \{a_i - b_i \mid i \in \mathbb{N}\} \subseteq \Sigma^*$$

Then P is a Church-Rosser system if and only if B is a Gröbner basis of $\langle B \rangle$. Furthermore, P is reduced if and only if B is reduced.

Proof. Assume first that B is a Gröbner basis. Let $D, E \in \Sigma^*$, D, E irreducible with respect to P . Then D and E are in normal form with respect to B (compare Definition 1.14), and hence $\text{NF}_{\leq}^{D_0}(D) = D$, $\text{NF}_{\leq}^{E_0}(E) = E$, as B is a Gröbner basis. Theorem 1.34 tells us that both D and E are the unique minimal element of their equivalence classes with respect to \sim_P . But by assumption $D \sim_P E$, by Theorem 3.22 we have $D = E$, and so they must coincide.

Now assume that P is a Church-Rosser system. Let $a \in \Sigma^*$ and assume a were irreducible with respect to P . Theorem 1.34 yields a D such that $a \sim_P D$. Theorem 3.22 implies that $a \sim_P D$, and after applying a finite number of reduction steps we obtain $a \rightarrow_P^* E$ with E irreducible, too. So $a \sim_P E$ and both are irreducible, by the Church-Rosser property we have $a = E$, which contradicts a irreducible. Hence there is a $a \in \Sigma^*$ with $\text{LM}(a) = a_0$ and $a \sim_P^* b$, and B is a Gröbner basis by characterization 1.18(a).

It is immediate from the definitions that $\langle B \rangle$ is reduced if and only if P is reduced.

Corollary 3.29. For every congruence relation \sim on Σ^* there exists a unique Church-Rosser system P with $\sim_P = \sim$.

Proof. There exists some commutative semigroup presentation P_0 with $\Sigma = \langle \Sigma, P_0 \rangle$ by Rédei's theorem A.7. Let $B = \{a_i - b_i \mid i \in \mathbb{N}\} \subseteq \Sigma^*$ and let B be the reduced Gröbner basis of

. Then $P = \{f^1; -A^0 j - i - A^2\}$ is a Church-Rosser system equivalent to P_0 . Uniqueness follows immediately from the uniqueness of reduced Gröbner bases.

We now translate some notation and results from the polynomial world to the land of Thue systems. Fix a commutative Thue system P .

Definition 3.30. For a word $D \in \Sigma^*$ let $NF_{P^1} D^0$ be the minimal word in $\gg D \gg_P$ with respect to \gg_P . We define the sets

$$\ast_P = \{D \in \Sigma^* \mid NF_{P^1} D^0 = D\} \quad \text{and} \quad \text{Min}_P = \{D \in \Sigma^* \mid D \in \ast_P\}.$$

We denote the minimal elements of a set S with respect to string containment by $\text{min } S$.

Lemma 3.31. Let P be a reduced Church-Rosser system.

- (i) \ast_P is the set of elements reducible with respect to P .
- (ii) $\text{Min}_P = \{f^1; j^1; -A^0 \in P\}$

Proof. As P is a Church-Rosser system, a string $D \in \Sigma^*$ is irreducible with respect to P if and only if it is minimal in its congruence class $\gg D \gg_P$, i. e. in \ast_P . This proves the first statement, the second statement is a direct translation of Theorem 1.32.

3.7 The size of a reduced Gröbner basis

In this section we restrict ourselves to a *degree-dominating* monomial order \gg with the property

$$|D| \leq |E| \Rightarrow D \gg E.$$

For example \gg_{grlex} and \gg_{grevlex} have this property, but \gg_{lex} does not.

Theorem 3.32 (Huynh 1986 [21]). For each $n \in \mathbb{N}$ there exists an ideal \mathfrak{I}_n generated by $O(1)^n$ differences of monomials of total degree $O(1)^n$ with the following properties:

Any Gröbner basis of \mathfrak{I}_n has at least 2^{2^n} elements and the maximal total degree of its elements is at least 2^{2^n} .

By Corollary 1.29 the equivalence of reduced Church-Rosser systems and reduced Gröbner bases of pure difference ideals, it suffices to prove the analogous statement for Church-Rosser systems:

Theorem 3.33. For each $n \in \mathbb{N}$ there exists a commutative Thue system R_n generated by $O(1)^n$ rules of length $O(1)^n$ with the following properties:

The reduced Church-Rosser system generating the same congruence as R_n has at least 2^{2^n} elements and the maximal length of one side of the rules is 2^{2^n} .

To prove this, we follow the strategy of Huynh [21], but we simplify the construction of a system $R_{=}$ with the following property:

Lemma 3.34. *Let $D = A(c_3 \ c_1 \ c_2 \ c_4 \ c_5)$ with $c_4 \ c_5 = e_{=}$. Then D is not minimal in $\gg D^0_{R_{=}}$ with respect to \gg , but any proper substring $D^0 \ j \ D$ is the minimal element of $\gg D^0_{R_{=}}$.*

We first show that this property implies the theorem.

Proof of Theorem 3.3. Consider the $e_{=} + 1$ distinct words $D: B A(c_3 \ c_1 \ c_2 \ c_4 \ c_5)$, $i = 0 \dots e_{=}$, each of length $e_{=} + 2$. By Lemma 3.34 each $D: 2 \ R_{=}$ is a *minimal* element of this set, and hence by Lemma 3.31 we have

$$\{D_0 \dots D_{e_{=}}\} \ \{j \ 1; \dots A^0 \ 2 \ P\} \ g$$

where P is the reduced Church-Rosser system equivalent to $R_{=}$. Hence $|P| \geq e_{=} + 1$ and P contains rules of length $\geq e_{=} + 1$.

We now construct $R_{=}$: Recall the construction of $P_{=} = P_{=}^0$ from section 3.4, then the alphabet is $\Sigma_{=} = \{A, A^0, A^{00}, c_1, c_2\}$ and $R_{=}$ consists of $P_{=}$ and the additional rules

$$\begin{aligned} A(c_3 \ c_1) &\rightarrow A(c_3 \ c_1 \ c_4) & A(c_3 \ c_2) &\rightarrow A(c_3 \ c_2 \ c_4) & (R) \\ A(c_3) &\rightarrow A^0(c_1) & & & (R_1^0) \\ A^0(c_1) &\rightarrow A^0(c_3) & & & (R_2^0) \\ A(c_3) &\rightarrow A^{00}(c_4) & & & (R_1^{00}) \\ A^{00}(c_4 \ c_2) &\rightarrow A^{00}(c_4 \ c_1) & & & (R_2^{00}) \end{aligned}$$

The rules and their intended behavior is visualized in figure 3.4. The proof of Lemma 3.34 makes use of the properties of $P_{=}$ from Theorem 3.13, and the additional fact that in the derivation $\gg_{\delta}^0(A^0(c_3)) \gg_{\delta}^{e_{=}}$ any intermediate expression has length ≥ 3 .

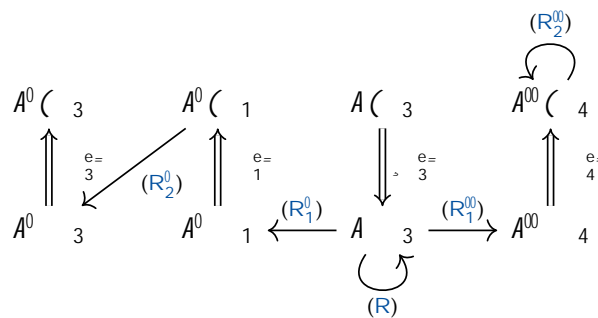


Figure 3.4: A visualization of the rules in $R_{=}$.

$\dim^1 A_{\leq d} = A$, for all $A \in \mathbb{Z}$, $A \geq 0$, which are generated by $O^1 = O$ polynomials of degrees bounded by 3 such that each Gröbner basis has a maximal degree of at least $3^{1-d} = A^{O(2^{1-d})}$ for any $d \geq 0$ and sufficiently large $3-A \in \mathbb{Z}$.

This shows that the upper bound from Theorem 2.17 is somewhat tight, in particular the degree of Gröbner bases may be loosely described as $2^{\Theta(2^{1-d})}$.

3.8 Hardness results of Gröbner bases

We finally prove some results on the complexity of Gröbner bases. Fix a monomial order which is degree-dominating.

Theorem 3.36. *Any algorithm which on input $\langle s_1, \dots, s_r \rangle$ computes the reduced Gröbner basis of $\langle s_i \rangle$ with respect to grlex uses in the worst case at least space $2^{\Theta(n)}$ and time $2^{2^{\Theta(n)}}$ in the length of n . This is independent of the base field \mathbb{K} , the encoding of field elements or whether binary or unary exponent representation of the monomials is used. This also holds true when restricting homogeneous ideals.*

Proof. Indeed, the polynomials from Theorem 3.32 are differences of monomials and hence use coefficients $\in \{0, -1\}$. Furthermore, the terms have bounded degree $\leq n$, so dense or sparse encoding of the monomials does not change the input length up to a constant factor.

The output consist of $2^{2^n} = e_n$ different polynomials, so the algorithm requires at least e_n steps and passes through at least e_n different configurations of the work tape. This implies that at least 2^n work tape cells have to be visited in order to represent e_n different internal configurations.

This lower bound matches the space and time requirements of Theorem 2.29 (for $\mathbb{K} = \mathbb{F}_2$).

Remark. We have seen that the size of the output of the Turing machine dictates a lower bound on the resources used by said machine. This is however not enough to prove that the *individual elements* of the output are hard to describe. For example, consider the function

$$f_j = 10^{-1-2-\dots-2^j} = 10^{-2^{j+1} + 1}$$

where numbers are encoded in binary. Then the length $|f_j|$ is exponential in j and hence double-exponential in $\lfloor \log_2 |f_j| \rfloor$, but on the other hand it is computationally trivial to decide whether $\langle f_j \rangle$ is part of the list $\{f_j\}$. This is *not* true in the case of the reduced Gröbner basis, as shown next.

Theorem 3.37 (GROEBM $_{\mathbb{K}}$ is EXPSPACE-hard). *The problem to decide whether a given polynomial f is a member of the reduced Gröbner basis of $\langle s_1, \dots, s_r \rangle$ is hard for EXPSPACE, regardless of the specified monomial order \prec .*

Proof. The reduction $EBC \stackrel{P}{m} CSG \stackrel{P}{m} IM_{\kappa}$ shows that IM_{κ} is EXPSPACE-hard even when restricting to the following subset \mathcal{S} :

- The polynomials $f_1 - f_2 - \dots - f_B$ are differences of monomials of positive degree.
- The polynomial f whose membership has to be decided is of the form $f = G^0 - H, G - H \in \mathcal{S}$.

After renumbering the variables we may assume that $G^0 = H$ for all $G^0 \in \mathcal{S}$, i.e. H is the smallest variable with respect to \prec . This implies that H is also the smallest monomial different from 1: Any monomial $<$ contains some variable G^0 , and hence $< \prec G^0 = H$. Moreover $1 \succ \langle$ for any monomial $< < 1$, since the f_i have positive degree (in Thue system language: No rule is applicable to the empty string).

Let $\mathcal{I} = \langle f_1 - \dots - f_B \rangle$ be an ideal generated by pure differences and \mathcal{G} the reduced Gröbner basis of \mathcal{I} . With this notation we claim

$$G \in \mathcal{S} \iff G \in \mathcal{G} \text{ if and only if } G \in \mathcal{H}.$$

If $G \in \mathcal{S}$, then surely $G \in \mathcal{H}$, as $\mathcal{S} \subseteq \mathcal{H}$. Conversely, assume $G \in \mathcal{H}$, then Theorem 1.34 shows that $NF \langle G \rangle = \langle H \rangle$ as $H \in \mathcal{G}$ and H is the smallest monomial in $\text{Mon}_{\mathcal{S}}$. The monomial G is minimally reducible with respect to \prec (Definition 1.31), as it is reducible ($NF \langle G \rangle = \langle H \rangle$), but its only proper divisor 1 is irreducible. Then Theorem 1.32 shows that $G \in \mathcal{S}$.

This proves $\stackrel{P}{m} \text{GROEBM}_{\kappa}$, where the reduction function is the identity map, so GROEBM_{κ} is EXPSPACE-hard, too.

Combining this with the exponential space algorithm for reduced Gröbner basis membership from chapter 2 yields the following completeness result:

Corollary 3.38. *The problem GROEBM_{κ} is EXPSPACE-complete.*

Remark. The hardness result of GROEBM_{κ} is implicitly contained in the literature, although not mentioned explicitly (to the knowledge of the author). The phrase “Gröbner bases are EXPSPACE-complete” mostly refers to the results of Theorem 2.29 and 3.36.

We close our discussion on the complexity of Gröbner bases with a recent result due to Rolnick & Spencer which shows that Gröbner bases are hard to approximate. Let $\epsilon \in (0, 1)$ be a constant and define the ϵ -Fractional Gröbner problem as follows:

- Input: $\langle f_1 - \dots - f_B \rangle \in \mathcal{S}$ polynomials
- Output: $\langle g_1 - \dots - g_r \rangle$, where $g_j = \sum_{i=1}^B \alpha_{ij} f_i$ with $\alpha_{ij} \in [0, 1]$ and $\langle g_1 - \dots - g_r \rangle$ is a Gröbner basis of $\langle f_1 - \dots - f_B \rangle$.

Theorem 3.39 (Rolnick & Spencer 2018 [46]). *For infinite fields \mathcal{K} , the ϵ -Fractional Gröbner problem is NP-hard for every fixed $\epsilon \in (0, 1)$. This is true for any monomial order and even when the polynomials are of degree ≤ 3 .*

The authors remark that this holds true even if polynomial time is measured not only in the size of the input, but also on the output [46, Theorem 3]. This is remarkable, since we know that Gröbner bases may become exceedingly large even for small inputs.

Conclusion

In this thesis we discussed the polynomial ideal membership and its connection to Gröbner bases. We have seen how Gröbner bases can be constructed using Buchberger's algorithm and how the multivariate division algorithm calculates normal forms which can be used to ideal membership. Although this approach is successful in many cases, especially with improved algorithms such as Faugère's F_5 algorithm, the worst-case complexity points to the limitations of Gröbner bases.

We have seen that both the ideal membership problem IM as well as the reduced Gröbner basis membership problem $GROEBM$ are EXPSpace-complete, in particular both problems are computationally very involved in their full generality. While it is intuitively clear that Gröbner bases must be at least as complex as the problem they are trying to solve (in an appropriate sense), the other results surveyed here suggest that they are sub-optimal in many important cases. Namely, the ideal membership problem for homogeneous polynomials IM_h is PSPACE-complete, but Gröbner basis computation for homogeneous ideals still requires exponential space in the worst case. Even worse, the number and degree of elements in a Gröbner basis of a homogeneous ideal may be double-exponential in the number of variables and generators of the ideal. Still, in many practical cases Gröbner bases are a good first choice for problems in computer algebra.

There are many other interesting topics related to the ideas presented here, we would like to point to three particular ones.

- When presented with a problem involving Gröbner basis computation, it is often difficult to estimate whether the computation will be a matter of minutes or months. Hence an important question is: What properties of an ideal make their Gröbner basis complicated? This is particularly important for applications in computational algebraic geometry, where many problems can be reduced to a Gröbner basis computation of some sort. We have seen that the number of variables and the dimension play a role both for the complexity of the ideal membership as well as the size of the reduced Gröbner basis. An important piece of the puzzle is the *regularity* of the ideal, a notion which is explored in the paper by Bayer & Mumford [3].
- Gröbner bases provide an approach to calculate normal forms of polynomials with respect to an ideal. If the system of polynomials has a finite number of solutions and is sufficiently general, then the set of solutions is closely connected to the quotient ring

$f = \frac{1}{4} \cdot s_1 \cdot \dots \cdot s_B$, for example its dimension over f is the number of solutions. Normal forms provide a way to compute effectively in this ring, see for example the *truncated normal form* by Telen [38]. For an analysis of the performance of several normal form algorithms see the work of Parkinson et. al [41].

- While symbolic computations on systems of polynomial equations are important, for many applications approximations to solutions are sufficient, especially for large systems. One of the most successful techniques is *homotopy continuation*, where a system of equations with known solutions is continuously transformed into the target system, and the solution paths are tracked. A particular implementation, which allows for certifying zeros (i. e. providing a certificate that a given approximate zero corresponds to an *actual* solution in f^{\approx}) is [HomotopyContinuation.jl](#) [6].



Appendix

A.1 Commutative Algebra

All rings in this thesis are assumed to be commutative and with unity. An ideal of a ring is an additive subgroup closed under multiplication with elements from R , i.e. $0 \in I$ and $r \in R, s \in I \implies rs \in I$. If S is a subset, then the ideal generated by S is

$$\langle S \rangle = \left\{ \sum_{i=1}^n r_i s_i \mid r_i \in R, s_i \in S \right\}$$

this is the smallest ideal containing S and S is a *generating set* of $\langle S \rangle$. If $I = \langle s_1, \dots, s_n \rangle$ then we write $I = \langle s_1, \dots, s_n \rangle$. An ideal of the form $\langle s \rangle$ is a *principal ideal*.

A ring R is *Noetherian* if one of the following equivalent conditions is satisfied [24, Thm. 2.9]

- (i) Any generating set of an ideal contains a finite generating set.
- (i') Every ideal of R admits a finite generating set.
- (ii) R satisfies the *ascending chain condition* on ideals: Any chain of ideals $I_0 \subseteq I_1 \subseteq I_2 \subseteq \dots$ eventually becomes stationary: $I_n = I_{n+1} = \dots$ for all $n \geq 0$.
- (iii) Any nonempty set of ideals in R has a maximal element with respect to inclusion.

The most important ring for our purposes is the ring of polynomials in a set of indeterminates over a field $K[x_1, \dots, x_n]$. A classical result asserts that this is a Noetherian ring.

Theorem A.1 (Hilbert basis theorem). *If R is a Noetherian ring, then so is $R[x_1, \dots, x_n]$. In particular $K[x_1, \dots, x_n]$ is a Noetherian ring.*

Proof. See for example [24, Thm. 2.11] or [19, Thm. 1.3.5]

If R and S are rings and $I \subseteq R, J \subseteq S$, then in general $\langle I \times J \rangle \subseteq \langle I \rangle \times \langle J \rangle$, but we have equality in the following special case:

Lemma A.2. *Let $K \subseteq L$ be two fields, $I = \langle s_1, \dots, s_n \rangle \subseteq K[x_1, \dots, x_n]$, then we have*

$$\langle I \rangle \subseteq K[x_1, \dots, x_n] \implies \langle I \rangle \otimes_K L = \langle I \rangle_L \subseteq L[x_1, \dots, x_n]$$

Proof. \mathbb{E} is a vector space over $\langle \cdot \rangle$, we can choose¹ a basis $\{1_{\beta} \mid \beta \in \mathbb{N}\}$ containing $1_{\beta_0} = 1 \in \mathbb{E}$. This then also constitutes a basis of the free module $\mathbb{E}\langle \cdot \rangle_{\mathbb{N}}$ over $\langle \cdot \rangle_{\mathbb{N}}$. Now consider an expression

$$s = \sum_{\beta \in \mathbb{N}} b_{\beta} \cdot 1_{\beta} \in \mathbb{E}\langle \cdot \rangle_{\mathbb{N}}.$$

We can write $b_{\beta} = \sum_{\gamma \in \mathbb{N}} b_{\beta\gamma} \cdot 1_{\gamma}$ with finitely many nonzero $b_{\beta\gamma} \in \langle \cdot \rangle_{\mathbb{N}}$, then the former equation becomes

$$s \cdot 1_{\beta_0} = s = \sum_{\beta \in \mathbb{N}} b_{\beta} \cdot 1_{\beta} = \sum_{\beta \in \mathbb{N}} \sum_{\gamma \in \mathbb{N}} b_{\beta\gamma} \cdot 1_{\beta} \cdot 1_{\gamma} = \sum_{\gamma \in \mathbb{N}} \left(\sum_{\beta \in \mathbb{N}} b_{\beta\gamma} \cdot 1_{\beta} \right) \cdot 1_{\gamma}.$$

Comparing the coefficient of the basis element $1_{\beta_0} = 1$ in this equation yields

$$s = \sum_{\beta \in \mathbb{N}} b_{\beta\beta_0} \cdot 1_{\beta} = \sum_{\beta \in \mathbb{N}} b_{\beta\beta_0} \cdot \langle \cdot \rangle_{\mathbb{N}}.$$

A.2 Commutative semigroups

A *semigroup* is a set together with an associative binary operation \cdot , as usual we will omit the operation, i. e. write 01 instead of $0 \cdot 1$. A *monoid* is a semigroup with a neutral element 1 ; one can always adjoin a neutral element to a semigroup if desired and since we will only consider monoids, we use these notions exchangeably. Our binary operation will be commutative in most interesting cases, hence we are really studying “commutative monoids”.

Example A.3 (Free (commutative) monoids). The most well-known semigroup in computer science is the *free monoid* over a finite alphabet Σ , consisting of strings of letters from Σ with string concatenation as the composition.

Similarly, the free commutative monoid over the alphabet $\Sigma = \{G_1, \dots, G_n\}$ is the set of (commuting) monomials

$$\text{Mon}^{\Sigma} = \{G_1^{z_1} \cdot G_2^{z_2} \cdot \dots \cdot G_n^{z_n} \mid z_i \in \mathbb{N}\}.$$

This semigroup is isomorphic to the commutative monoid of integer vectors \mathbb{N}^n .

A homomorphism of semigroups is a map of sets $f: S \rightarrow T$ with $f(st) = f(s)f(t)$ for $s, t \in S$, in the case of monoids we also require that $f(1) = 1$.

Remark. Notice that the concept of “kernel of a homomorphism” from group theory is not well-behaved in this general situation. For once, the kernel of $f: S \rightarrow T$ is usually defined as $f^{-1}(1)$, and S needn’t have a neutral element. And even if both S and T have neutral elements, then we may have $f^{-1}(1) = \emptyset$ even though f is *not* injective! Consider for

¹This can be easily done for finite extension of $\langle \cdot \rangle$, and is *always* possible assuming the axiom of choice.

example

$$f: \mathbb{Z}^2 \rightarrow \mathbb{Z} \times \mathbb{Z} \quad f(a_1, a_2) = (a_1, a_2)$$

then $f^{-1}(0) = \{(0, 0)\}$, but in general $f^{-1}(z)$ has precisely $|z| + 1$ preimages

$$f^{-1}(z) = \{(0, z), (-1, z+1), \dots, (z, 1)\} \quad \mathbb{Z}^2.$$

The previous example show that the naive definition of a kernel fails to measure the number of preimages in any meaningful way. But we can still define the following:

Definition A.4 (Congruence relation). A congruence relation on a semigroup S is an equivalence relation (i. e. reflexive, symmetric and transitive) such that for $a, b, c \in S$

$$a \equiv b \text{ and } c \equiv d \Rightarrow ac \equiv bd. \tag{y}$$

If $f: S \rightarrow T$ is a homomorphism of semigroups, then

$$a \equiv b \text{ in } S \Rightarrow f(a) \equiv f(b) \text{ in } T$$

defines a congruence relation on T . Conversely, if \equiv is a congruence relation on T , then the set of equivalence classes T/\equiv forms a semigroup (with $(a \equiv b) \cdot (c \equiv d) \equiv (ac) \equiv (bd)$) and the surjective homomorphism

$$f: S \rightarrow T/\equiv \quad f(a) = a \equiv$$

defines the same equivalence relation $a \equiv b \iff f(a) = f(b)$.

Definition A.5 (Commutative semigroup presentation). A semigroup presentation is a pair $\langle S, R \rangle$, where S is a set of generators and R is a set of relations. The congruence relation \equiv_R generated by R is the smallest congruence relation on S^* containing R . The commutative semigroup $\langle S \mid R \rangle$ is the quotient of S^* / \equiv_R .

The presentation is finite if both S and R are finite. A commutative semigroup is finitely presented if it is isomorphic to $\langle S \mid R \rangle$ for some finite presentation. y

Lemma A.6. If R is a set of relations, then $G \equiv_R H$ if and only if there is a (possibly empty) sequence $G = G_0 \cdot G_1 \cdot \dots \cdot G_n = H$ such that

$$G_{i+1} = g_i \cdot g_i^{-1} \cdot G_i \text{ or } G_{i+1} = g_i \cdot g_i$$

for suitable $g_i \in S^*$ and $g_i^{-1} \cdot g_i$ or $g_i \cdot g_i$ in R .

Proof. \equiv_R contains R , by symmetry all $g_i^{-1} \cdot g_i$ for $g_i \in S^*$ and by reflexivity all $g_i \cdot g_i$, $g_i \in S^*$. Since it is a congruence relation, it contains all pairs $g_i \cdot g_i^{-1} \cdot G_i \equiv g_i \cdot g_i \cdot G_i$ and hence by transitivity also $G_i \equiv G_{i+1}$ as in the lemma.

Conversely, it is easy to see that the relation

$$G \equiv H \text{ if there exists a sequence as above}$$

is already a congruence relation, so the two relations coincide.

It turns out that every finitely generated commutative monoid is in fact finitely presented!

Theorem A.7 (Rédei). *If M is finitely generated, i. e. we have a surjective homomorphism $\phi: F \rightarrow M$ for a finite set S , then M is generated by finitely many relations.*

We can sketch a elegant and short proof due to Freyd [18]. If M is a monoid, then its monoid ring $R = \mathbb{k}[M]$ is the \mathbb{k} -algebra whose underlying vector space has $\{x^g\}_{g \in M}$ as a basis and the multiplication is defined by $x^g x^h = x^{gh}$ on this basis and extended by bilinearity. For example, $R = \mathbb{k}[G_1, \dots, G_n]$ is just the polynomial ring in n variables.

Proof (sketch). Let $M = \langle S \rangle$. Assume that M is not finitely generated, then there exists an infinitely ascending sequence $R_1 \subset R_2 \subset \dots$ contained in R such that $R_\ell \subset R_{\ell+1}$ for all ℓ . Let $\phi_\ell: R \rightarrow R_\ell$, then this gives rise to a chain of surjective but non-bijective homomorphisms

$$R \xrightarrow{\phi_1} R_1 \xrightarrow{\phi_2} R_2 \xrightarrow{\phi_3} \dots$$

These maps yield surjective ring homomorphisms $R \rightarrow R_\ell$, let I_ℓ be the kernel. By construction $I_1 \subset I_2 \subset \dots$ is a strictly increasing chain of ideals in $R = \mathbb{k}[x_1, \dots, x_n]$, but this contradicts the Noetherian property of polynomial rings (Hilbert's basis theorem A.1)!

Index

- Fractional Gröbner problem,
- Church-Rosser system
 - reduced,
- church-rosser system,
- Counter machine
 - computation bounded by =,
- counter machine,
- degree-dominating,
- dehomogenization,
- dimension
 - of an ideal,
- exponentially bounded counter machines,
- free commutative monoid,
- generating set,
- Gröbner basis,
 - membership problem,
- graded lexicographic ordering,
- graded reverse lexicographic order,
- Gröbner basis
 - interreduced,
 - reduced,
- homogenization,
- Ideal membership problem,
- initial ideal,
- leading coefficient,
- leading monomial,
- leading term, ,
- level
 - of a symbol,
 - of a word,
- lexicographic ordering,
- monomial,
 - minimally reducible,
- monomial ideal,
- monomial ordering,
- multidegree,
- noetherian,
- normal forms,
- normalized,
- Nullstellensatz,
- parallel computation thesis,
- polynomial,
 - degree,
 - reducible w.r.t. an ideal,
- principal ideal,
- pure binomials,
- reducible
 - word,
- s-polynomial,
- semigroup,
- stack machine,
- string rewriting system,
- support
 - of a polynomial,
- Thue system

commutative,
equivalent,
thue system,

Word problem for commutative semi-
groups,

List of Symbols

\mathbb{F}_q	the finite field of order q (unique up to isomorphism)
$\mathbb{Z}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	the natural numbers (with 0), the integers, the rational numbers, the real numbers and the complex numbers
\underline{x}	a set of variables x_1, \dots, x_n
$\text{Mon}_{\underline{x}}$	the set of monomials in n variables and in \mathbb{C}
$\text{supp}^1 f$	the monomials occurring in a polynomial f
$\text{IM}_{\langle \mathcal{I} \rangle}$	the ideal membership problem over the field \mathbb{C}
$\text{HNST}_{\langle \mathcal{I} \rangle}$	the decision problem to Hilbert's Nullstellensatz
\prec, \prec_P	a monomial order and the induced order on finite sets
$\prec_{\text{lex}}, \prec_{\text{grlex}}, \prec_{\text{grevlex}}$	the lexicographic, graded lexicographic and graded reverse lexicographic order
$\text{LT}^1 f, \text{LM}^1 f, \text{LC}^1 f$	the leading term, leading monomial and leading coefficient of a polynomial f
$\text{mdeg}^1 f$	the multidegree of a polynomial f
$\text{rem}^1 f; \delta_1, \dots, \delta_n$	the remainder of f with respect to $\delta_1, \dots, \delta_n$
$\text{IN}^1 \mathcal{I}$	the initial ideal generated by leading terms of \mathcal{I}
$\text{NF}^1 f, \text{NF}^1 \mathcal{I}$	the set of normal forms of f with respect to a set \mathcal{I} or an ideal
$\text{GROEBM}_{\langle \mathcal{I} \rangle}$	the membership problem for reduced Gröbner bases of polynomials over \mathbb{C}
$\sim_{\langle \mathcal{I} \rangle}$	the congruence relation defined by the ideal \mathcal{I} and the equivalence class of f
\mathbf{w}	a weight matrix and the associated monomial order
$\text{gcd}^1 m, n; \text{lcm}^1 m, n$	the greatest common divisor and least common multiple of two monomials
$\text{Spoly}^1 f, g$	the S-polynomial of f, g
$h_{f, a}$	the homogenization of f and dehomogenization of h
P, P', P''	a Thue system and the derivation relations
Σ^*	the set of commutative words over Σ
\sim, \sim^*	the configuration transition relation for a (counter) machine and its transitive reflexive closure

$\text{rep}^1 @ - 2_1 - 2_2 - 2_3^0$	the string representation of a configuration of a counter machine
$^1 = - P =^0$	the commutative Thue system by Mayr & Meyer counting to 2^2
$\text{LVL}^1 A^0$	the set of words in $=$ of level A
$\text{BOX}^1 G^0$	the box of G of the form $^1 (j^{01} \ 1 \cdots - \ 4^0$
$G^1 P^0$	the graph associated to a commutative Thue system P
$*_{P-} P$	the set of irreducible and reducible words with respect to a commutative Thue system P

List of Definitions and Theorems

1.3	Definition (Polynomial)	5
1.4	Definition (Ideal membership problem, IM_{κ})	6
1.8	Theorem (Hilbert's Nullstellensatz)	7
1.9	Definition (Nullstellensatz, $HNST_{\kappa}$)	7
1.10	Definition (Monomial ordering)	9
1.14	Definition (Normal form, NF)	12
1.18	Theorem (Characterizations of Gröbner bases)	13
1.19	Definition (Gröbner basis)	13
1.22	Theorem (The normal form map NF)	14
1.24	Definition (Interreduced Gröbner basis)	15
1.27	Definition (Reduced Gröbner basis)	16
1.30	Definition (Reduced Gröbner basis membership problem, $GROEBM_{\kappa}$)	17
1.31	Definition ((Ir)reducible, minimally reducible)	17
1.34	Theorem (Kopenhagen & Mayr 1999 [27])	19
1.36	Theorem (Robbiano 1985 [44])	21
2.1	Definition (S-Polynomial)	23
2.3	Theorem (Buchberger's criterion)	24
2.5	Theorem (Correctness of Buchberger's algorithm)	26
2.10	Theorem (Hermann 1926 [20])	29
2.11	Theorem (Kóllar 1988 [26])	29
2.12	Definition (Dimension of an ideal)	30
2.13	Theorem (Dickenstein et al. 1991 [12])	30
2.14	Definition ((De)homogenization)	30
2.16	Theorem (Dubé 1990 [14])	31
2.17	Theorem (Mayr & Ritscher 2013 [34])	31
2.21	Theorem (Fortune & Wyllie 1978 [17])	35
2.22	Theorem (Csanky 1976 [11])	35
2.23	Theorem (Ibarra, Moran & Rosier 1980 [22])	35
2.24	Theorem (Mayr 1989 [31], $IM_{\kappa} \in 2 \text{ EXPSPACE}$)	36
2.27	Theorem ($IM_{h,\kappa} \in HNST_{\kappa} \in 2 \text{ PSPACE}$)	36
2.28	Theorem (Koiran 1996 [25])	37
2.31	Theorem (Krick & Logar 1991 [29])	39

3.1	Definition (String rewriting system, Thue system)	42
3.3	Definition (, Commutative Thue system)	43
3.5	Definition (Word problem for commutative semigroups, CSG)	43
3.7	Theorem (CSG_h is PSPACE-hard)	45
3.8	Definition (Counter machine)	46
3.11	Theorem (EBC is EXPSPACE-complete)	49
3.13	Theorem (Mayr & Meyer, 1982 [33])	50
3.16	Theorem (CSG is EXPSPACE-hard)	52
3.17	Definition (Level of a word, box)	54
3.20	Theorem (Bayer & Stillman [4])	56
3.23	Theorem (CSG reduces to IM_ℓ)	61
3.25	Theorem (Bürgisser 1998 [9])	62
3.26	Definition ((Ir)reducible words)	62
3.27	Definition (Church-Rosser system, reduced)	63
3.32	Theorem (Huynh 1986 [21])	64
3.37	Theorem ($GROEBM_\ell$ is EXPSPACE-hard)	67
3.39	Theorem (Rolnick & Spencer 2018 [46])	68
A.1	Theorem (Hilbert basis theorem)	73
A.4	Definition (Congruence relation)	75
A.5	Definition (Commutative semigroup presentation)	75
A.7	Theorem (Rédei)	76

List of Figures

1.1	The perpendicular bisectors of a triangle intersecting in the circumcenter " . . .	4
3.1	The chain of complexity-theoretic reductions.	41
3.2	How to represent the tape of a Turing machine with two stacks.	48
3.3	Rules (A)–(G),(V) in the context of boxes.	55
3.4	A visualization of the rules in $\mathcal{R}_=$	65

Bibliography

- [1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge: Cambridge University Press, 2009. ISBN: 9780521424264. DOI: [10. 1017/CB09780511804090](https://doi.org/10.1017/CB09780511804090) (cit. on pp. [37](#), [41](#)).
- [2] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. “On the complexity of the F5 Gröbner basis algorithm”. In: *Journal of Symbolic Computation* 70 (2015), pp. 49–70. ISSN: 0747-7171. DOI: <https://doi.org/10.1016/j.jsc.2014.09.025> (cit. on p. [28](#)).
- [3] Dave Bayer and David Mumford. *What can be computed in algebraic geometry?* 1993. DOI: [10. 48550/ARXIV.ALG-GEOM/9304003](https://doi.org/10.48550/ARXIV.ALG-GEOM/9304003) (cit. on p. [71](#)).
- [4] David Bayer and Michael Stillman. “On the complexity of computing syzygies”. In: *Journal of Symbolic Computation* 6.2 (1988), pp. 135–147. ISSN: 0747-7171. DOI: [https://doi.org/10.1016/S0747-7171\(88\)80039-7](https://doi.org/10.1016/S0747-7171(88)80039-7) (cit. on pp. [54](#), [56](#), [60](#), [62](#)).
- [5] Lenore Blum et al. *Complexity and Real Computation*. Springer New York, Oct. 2012. 472 pp. ISBN: 1461268737. DOI: [10. 1007/978-1-4612-0701-6](https://doi.org/10.1007/978-1-4612-0701-6) (cit. on p. [7](#)).
- [6] Paul Breiding and Sascha Timme. “HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia”. In: *Mathematical Software – ICMS* . Ed. by James H. Davenport et al. Cham: Springer International Publishing, 2018, pp. 458–465. ISBN: 978-3-319-96418-8 (cit. on p. [72](#)).
- [7] Paul Breiding et al. *Nonlinear Algebra and Applications*. 2021. DOI: [10. 48550/ARXIV. 2103. 16300](https://doi.org/10.48550/ARXIV.2103.16300) (cit. on p. [4](#)).
- [8] Bruno Buchberger. “Bruno Buchberger’s PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal”. In: *Journal of Symbolic Computation* 41.3 (2006). Logic, Mathematics and Computer Science: Interactions in honor of Bruno Buchberger (60th birthday), pp. 475–511. ISSN: 0747-7171. DOI: <https://doi.org/10.1016/j.jsc.2005.09.007> (cit. on p. [26](#)).
- [9] Peter Bürgisser. “On the Parallel Complexity of the Polynomial Ideal Membership Problem”. In: *Journal of Complexity* 14.2 (1998), pp. 176–189. ISSN: 0885-064X. DOI: <https://doi.org/10.1006/j.com.1998.0472> (cit. on p. [62](#)).
- [10] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. “Using the Groebner basis algorithm to find proofs of unsatisfiability”. In: *Proceedings of STOC’* (Mar. 2000). DOI: [10. 1145/237814. 237860](https://doi.org/10.1145/237814.237860) (cit. on p. [27](#)).

- [11] Laszlo Csanky. “Fast Parallel Matrix Inversion Algorithms”. In: *SIAM Journal on Computing* 5.4 (1976), pp. 618–623. DOI: [10. 1137/0205040](https://doi.org/10.1137/0205040) (cit. on p. 35).
- [12] Alicia Dickenstein et al. “The membership problem for unmixed polynomial ideals is solvable in single exponential time”. In: *Discrete Applied Mathematics* 33.1 (1991), pp. 73–94. ISSN: 0166-218X. DOI: [https://doi.org/10.1016/0166-218X\(91\)90109-A](https://doi.org/10.1016/0166-218X(91)90109-A) (cit. on p. 30).
- [13] Thomas Dubé, B. Mishra, and Chee-Keng Yap. “Admissible Orderings and Bounds for Gröbner Basis Normal Form Algorithms”. In: (1986) (cit. on pp. 21, 33).
- [14] Thomas W. Dubé. “The Structure of Polynomial Ideals and Gröbner Bases”. In: *SIAM Journal on Computing* 19.4 (Aug. 1990), pp. 750–773. DOI: [10. 1137/0219053](https://doi.org/10.1137/0219053) (cit. on p. 31).
- [15] David Eisenbud and Bernd Sturmfels. “Binomial Ideals”. In: *Duke Math. J.* 84 (Feb. 1994). DOI: [10. 1215/S0012-7094-96-08401-X](https://doi.org/10.1215/S0012-7094-96-08401-X) (cit. on p. 18).
- [16] Jean Charles Faugère. “A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)”. In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation. ISSAC '02*. Lille, France: Association for Computing Machinery, 2002, pp. 75–83. ISBN: 1581134843. DOI: [10. 1145/780506. 780516](https://doi.org/10.1145/780506.780516) (cit. on p. 28).
- [17] Steven Fortune and James Wyllie. “Parallelism in Random Access Machines”. In: *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. STOC '78*. San Diego, California, USA: Association for Computing Machinery, 1978, pp. 114–118. ISBN: 9781450374378. DOI: [10. 1145/800133. 804339](https://doi.org/10.1145/800133.804339) (cit. on p. 35).
- [18] Peter J. Freyd. “Rédei’s finiteness theorem for commutative semigroups”. In: 1968. DOI: [10. 1090/S0002-9939-1968-0227290-4](https://doi.org/10.1090/S0002-9939-1968-0227290-4) (cit. on p. 76).
- [19] Gert-Martin Greuel and Gerhard Pfister. *A Singular Introduction to Commutative Algebra*. Springer Berlin Heidelberg, Nov. 2007. 712 pp. ISBN: 3540735410. DOI: [10. 1007/978-3-540-73542-7](https://doi.org/10.1007/978-3-540-73542-7) (cit. on pp. 3, 7, 13, 28, 30, 73).
- [20] Grete Hermann. “Die Frage der endlich vielen Schritte in der Theorie der Polynomideale”. In: *Mathematische Annalen* 95 (1926), pp. 736–788 (cit. on p. 29).
- [21] Dung T. Huynh. “A Superexponential Lower Bound for Gröbner Bases and Church-Rosser Commutative Thue Systems”. In: *Inf. Control.* 68 (1986), pp. 196–206 (cit. on pp. 41, 62, 64, 65).
- [22] Oscar H. Ibarra, Shlomo Moran, and Louis E. Rosier. “A note on the parallel complexity of computing the rank of order n matrices”. In: *Information Processing Letters* 11.4,5 (1980) (cit. on p. 35).
- [23] Jürgen Gerhard Joachim von zur Gathen. *Modern Computer Algebra*. Cambridge University Press, Mar. 2017. 812 pp. ISBN: 1107039037. DOI: [10. 1017/CB09781139856065](https://doi.org/10.1017/CB09781139856065) (cit. on pp. 3, 4, 9).

- [24] Gregor Kemper. *A course in commutative algebra*. Springer, 2011, p. 246. ISBN: 9783642035456 (cit. on pp. [1](#), [3](#), [7](#), [13](#), [23](#), [24](#), [73](#)).
- [25] Pascal Koiran. “Hilbert’s Nullstellensatz Is in the Polynomial Hierarchy”. In: *Journal of Complexity* 12.4 (1996), pp. 273–286. ISSN: 0885-064X. DOI: <https://doi.org/10.1006/j.com.1996.0019> (cit. on p. [37](#)).
- [26] Janos Kollar. “Sharp Effective Nullstellensatz”. In: *Journal of the American Mathematical Society* 1.4 (1988), pp. 963–975. ISSN: 08940347, 10886834. URL: <http://www.jstor.org/stable/1990996> (cit. on p. [29](#)).
- [27] Ulla Koppenhagen and Ernst W. Mayr. “An Optimal Algorithm for Constructing the Reduced Gröbner Basis of Binomial Ideals”. In: *Journal of Symbolic Computation* 28.3 (1999), pp. 317–338. ISSN: 0747-7171. DOI: <https://doi.org/10.1006/j.sco.1999.0285> (cit. on pp. [18](#), [19](#), [39](#)).
- [28] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra*. Springer Berlin Heidelberg, 2000. DOI: [10.1007/978-3-540-70628-1](https://doi.org/10.1007/978-3-540-70628-1) (cit. on pp. [3](#), [14](#)).
- [29] Teresa Krick and Alessandro Logar. “Membership problem, Representation problem and the Computation of the Radical for one-dimensional Ideals”. In: *Effective Methods in Algebraic Geometry*. Ed. by Teo Mora and Carlo Traverso. Boston, MA: Birkhäuser Boston, 1991, pp. 203–216. ISBN: 978-1-4612-0441-1. DOI: [10.1007/978-1-4612-0441-1_13](https://doi.org/10.1007/978-1-4612-0441-1_13) (cit. on p. [39](#)).
- [30] Klaus Kühnle and Ernst W. Mayr. “Exponential Space Computation of Gröbner Bases”. In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation*. ISSAC '96. Zurich, Switzerland: Association for Computing Machinery, 1996, pp. 63–71. ISBN: 0897917960. DOI: [10.1145/236869.236900](https://doi.org/10.1145/236869.236900) (cit. on pp. [17](#), [33](#)).
- [31] Ernst W. Mayr. “Membership in polynomial ideals over \mathbb{Q} is exponential space complete”. In: *STACS*. Ed. by B. Monien and R. Cori. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 400–406. ISBN: 978-3-540-46098-5 (cit. on p. [36](#)).
- [32] Ernst W. Mayr. “Some Complexity Results for Polynomial Ideals”. In: *Journal of Complexity* 13.3 (1997), pp. 303–325. ISSN: 0885-064X. DOI: [10.1006/j.com.1997.0447](https://doi.org/10.1006/j.com.1997.0447) (cit. on pp. [23](#), [28](#), [41](#), [44](#)).
- [33] Ernst W. Mayr and Albert R. Meyer. “The complexity of the word problems for commutative semigroups and polynomial ideals”. In: *Advances in Mathematics* 46.3 (Dec. 1982), pp. 305–329. DOI: [10.1016/0001-8708\(82\)90048-2](https://doi.org/10.1016/0001-8708(82)90048-2) (cit. on pp. [29](#), [41](#), [50](#)).
- [34] Ernst W. Mayr and Stephan Ritscher. “Dimension-dependent bounds for Gröbner bases of polynomial ideals”. In: *Journal of Symbolic Computation* 49 (2013). The International Symposium on Symbolic and Algebraic Computation, pp. 78–94. ISSN: 0747-7171. DOI: doi.org/10.1016/j.jsc.2011.12.018 (cit. on pp. [30](#), [31](#)).

- [35] Ernst W. Mayr and Stefan Toman. “Complexity of Membership Problems of Different Types of Polynomial Ideals”. In: *Algorithmic and Experimental Methods in Algebra, Geometry, and Number Theory*. Ed. by Gebhard Böckle, Wolfram Decker, and Gunter Malle. Cham: Springer International Publishing, 2017, pp. 481–493. ISBN: 978-3-319-70566-8. DOI: [10.1007/978-3-319-70566-8_20](https://doi.org/10.1007/978-3-319-70566-8_20) (cit. on pp. 23, 28, 37).
- [36] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. USA: Prentice-Hall, Inc., 1967. ISBN: 0131655639 (cit. on pp. 46, 49).
- [37] H. Michael Möller and Ferdinando Mora. “Upper and lower bounds for the degree of Groebner bases”. In: *EUROSAM* . Ed. by John Fitch. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 172–183. ISBN: 978-3-540-38893-7 (cit. on p. 30).
- [38] Bernard Mourrain, Simon Telen, and Marc Van Barel. *Truncated Normal Forms for Solving Polynomial Systems: Generalized and Efficient Algorithms*. 2018. DOI: [10.48550/ARXIV.1803.07974](https://doi.org/10.48550/ARXIV.1803.07974) (cit. on p. 72).
- [39] Victor Pan. “Complexity of parallel matrix computations”. In: *Theoretical Computer Science* 54.1 (1987), pp. 65–85. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/0304-3975\(87\)90019-3](https://doi.org/10.1016/0304-3975(87)90019-3) (cit. on p. 35).
- [40] Behrooz Parhami. *Introduction to Parallel Processing: Algorithms and Architectures*. Boston, MA: Springer US, 2002. ISBN: 978-0-306-46964-0. DOI: [10.1007/0-306-46964-2_5](https://doi.org/10.1007/0-306-46964-2_5) (cit. on p. 34).
- [41] Suzanna Parkinson et al. *Analysis of Normal-Form Algorithms for Solving Systems of Polynomial Equations*. 2021. DOI: [10.48550/ARXIV.2104.03526](https://doi.org/10.48550/ARXIV.2104.03526) (cit. on p. 72).
- [42] Emil L. Post. “Recursive Unsolvability of a problem of Thue”. In: *Journal of Symbolic Logic* 12.1 (1947), pp. 1–11. DOI: [10.2307/2267170](https://doi.org/10.2307/2267170) (cit. on p. 42).
- [43] Stephan Ritscher. “Degree Bounds and Complexity of Gröbner Bases of Important Classes of Polynomial Ideals”. PhD thesis. Technische Universität München, 2012 (cit. on pp. 15, 20, 34, 36, 38).
- [44] Lorenzo Robbiano. “Term orderings on the polynomial ring”. In: *EUROCAL '85* . Ed. by Bob F. Caviness. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 513–517. ISBN: 978-3-540-39685-7 (cit. on p. 21).
- [45] J. Maurice Rojas. “Dedekind Zeta Functions and the Complexity of Hilbert’s Nullstellensatz”. In: *arXiv: Number Theory* (2003) (cit. on p. 37).
- [46] David Rolnick and Gwen Spencer. “On the robust hardness of Gröbner basis computation”. In: *Journal of Pure and Applied Algebra* 223.5 (2019), pp. 2080–2100. ISSN: 0022-4049. DOI: <https://doi.org/10.1016/j.jpaa.2018.08.016> (cit. on pp. 68, 69).

-
- [47] Chee K. Yap. “A new lower bound construction for commutative thue systems with applications”. In: *Journal of Symbolic Computation* 12.1 (1991), pp. 1–27. ISSN: 0747-7171. DOI: [https://doi.org/10.1016/S0747-7171\(08\)80138-1](https://doi.org/10.1016/S0747-7171(08)80138-1) (cit. on p. 60).