Functions and Polynomials over Finite Groups from the Computational Perspective

Gábor Horváth

A thesis submitted to the University of Hertfordshire in partial fulfilment of the requirements of the degree of

Doctor of Philosophy

The programme of research was carried out in the School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire.

March, 2008

Acknowledgements

I am grateful to my supervisors Chrystopher Nehaniv and Csaba Szabó. They always provided me guidance when I felt lost.

I thank every mathematician with whom my research ideas were discussed.

I received much support and encouragement from my family, especially on those days when I felt my whole world collapsing.

Finally I would like to thank everyone who directly or indirectly helped me to finish this thesis.

Abstract

In the thesis we investigate the connections between arbitrary functions and their realizing polynomials over finite algebras. We study functionally complete algebras, i.e. algebras over which every function can be realized by a polynomial expression. We characterize functional completeness by the so called Stone–Weierstrass property, and we determine the functionally complete semigroups and semirings. Then we investigate the computational perspective of the function–polynomial relationships over finite groups. We consider the efficient representability, the equivalence, and the equation solvability problems.

We approach the efficient representability problem from three directions. We consider the length of functions, we investigate the circuit complexity of functions, and we analyse the finite-state sequential machine representation of Boolean functions. From each of these viewpoints we give bounds on the potential efficiency of computations based on functionally complete groups compared to computations based on the two-element Boolean algebra.

Neither the equivalence problem nor the equation solvability problem has been completely characterized for finite groups. The complexity of the equivalence problem was only known for nilpotent groups. In the thesis we determine the complexity of the equivalence problem for certain meta-Abelian groups and for all non-solvable groups.

The complexity of the equation solvability problem is known for nilpotent groups and for non-solvable groups. There are no results about the complexity of the equation solvability problem for solvable non-nilpotent groups apart from the case of certain meta-cyclic groups that we present in the thesis. Moreover, we determine the complexity of the equation solvability problem for all functionally complete algebras.

The idea of the extended equivalence problem emerges from the observation that the commutator might significantly change the length of group-polynomials. We characterize the complexity of the extended equivalence problem for finite groups. For many finite groups we determine the complexity of the equivalence problem if the commutator is considered as the basic operation of the group.

Contents

Chapter 1

Introduction

Nowadays, computers play larger and larger role in everyday life and in scientific research. This is especially true in mathematics and in algebra, where one often wants to perform calculations or computations with a machine. Computers are based on the two-element Boolean algebra, namely $\mathbf{B} = (\{0,1\}, \neg, \vee, \wedge)$, where $\neg(0) = 1$, $\neg(1) = 0$, $\vee(0,0) = 0$, $\vee(0,1) = \vee(1,0) = \vee(1,1) = 1$, $\wedge(0,0) = \wedge(1,0) = \wedge(0,1) = 0$ and $\wedge(1,1) = 1$. Instead of $\vee(x,y)$ we write $x \vee y$ and instead of $\wedge(x,y)$ we use $x \wedge y$. The algebra $\mathbf{B}$ has a special property which makes the computers universal, namely every arbitrary function from $\{0,1\}^n$ to $\{0,1\}$ can be expressed by the basic operations $\neg, \vee$ and $\wedge$. This property is called *functional completeness*. However, not only $\mathbf{B}$ has this property.

By a *functionally complete* algebra $\mathbf{A}$ we mean an algebra with underlying set A and with basic operations $f_1, \ldots, f_m$ such that for every nonnegative integer n and for every function $f\colon A^n \to A$ there is a polynomial expression $p(x_1, \ldots, x_n)$ over $\mathbf{A}$ such that for every n-tuple $(a_1, \ldots, a_n) \in A^n$ we have $p(a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$. (*Polynomial expressions* are expressions built up from variables, constants from $\mathbf{A}$ and the basic operations of $\mathbf{A}$ using composition.) The two-element Boolean algebra, matrix rings over finite fields, and the finite simple non-Abelian groups are examples for functionally complete algebras [40, 41, 26]. A computer based on any of these algebras offers an alternative paradigm for computation.

To assess the power of other functionally complete algebras (especially groups) for providing a basis for computer science, we investigate the connections between functions and their representing polynomials. An arbitrary function can be represented by many polynomials and in many ways. Usually these polynomials are required to satisfy some natural conditions, such as shortness or efficient computability. In other cases we are given polynomials, and we are interested in whether the functions represented by the polyno-

mials have some common properties, such as: are the functions equal or do they attain the same value for some substitution? This work investigates these problems mainly over finite groups and therefore consists of three main themes.

1. Find representing polynomials for an arbitrary function over a given finite functionally complete group. We are especially interested in those representing polynomials which are either short or fast computable. This problem is the *efficient representability problem.*

2. Decide whether or not two polynomials represent the same function over a given finite group. We are especially interested in the computational complexity of this question in the length of the two polynomials. This problem is called the *equivalence problem.*

3. Decide whether two functions, which are represented by two polynomials over a given finite group, attain the same value at some substitution. We are especially interested in the computational complexity of this question in the length of the two polynomials. This problem is called the *equation solvability problem.*

In Chapters 2, 3 and 4 we are interested mainly in the first theme, while Chapters 5, 6, 7 and 8 focus on the latter two themes, which are closely related.

Now we give a brief survey on all three themes by recalling their background. Then we explain how the Chapters of the thesis relate to the former results. At the end of this Chapter we summarize the different methods and their importance.

1.1 The efficient representability problem

A natural question to ask is how a function can be represented as a polynomial. More interestingly, whether there is a short way of representing and a fast way of computing an arbitrary or a specific function over a given functionally complete algebra. These questions have been thoroughly investigated before for the two-element Boolean algebra (see e.g. [40]) or for rings (see e.g. [29]), but there are very few results for groups. Surprisingly, the original paper [26], characterizing the functionally complete groups, is not algorithmic: Maurer and Rhodes first prove that a finite group $\mathbf{G}$ has the so-called Stone–Weierstrass property if and only if it is simple and non-Abelian. Then they prove for groups that functional completeness follows

from the Stone–Weierstrass property. In the thesis we are particularly interested in functionally complete groups. We note that some of our results apply in a more general context, e.g. we prove theorems which hold for every functionally complete algebra.

In Chapter 2 we first give a basic overview about functionally complete algebras.

Definition. (equivalent to Definition 2) Let $\mathbf{A}$ be a finite algebra and let S be a finite nonempty set. Let $\mathbf{F}$ be an arbitrary subalgebra of $\mathbf{A}^S$, such that:

1. $\mathbf{F}$ contains the constant functions, namely for every $a \in \mathbf{A}$ there is a function $f_a \in \mathbf{F}$ such that for every $s \in S$ we have $f_a(s) = a$.

2. $\mathbf{F}$ separates every two elements of S, namely for every $s_1 \neq s_2 \in S$ there exists a function $f \in \mathbf{F}$ such that $f(s_1) \neq f(s_2)$.

If for every S these two properties imply that $\mathbf{F} = \mathbf{A}^S$, then we say that $\mathbf{A}$ has the *Stone–Weierstrass property.*

We prove that the Stone–Weierstrass property is equivalent with the functional completeness for any finite algebra, not only for a group:

Theorem. *(Theorem 3). Let* $\mathbf{A}$ *be a finite algebra. Then* $\mathbf{A}$ *has the Stone–Weierstrass property if and only if* $\mathbf{A}$ *is functionally complete.*

Then we determine the functionally complete classical algebras. Theorem 14 in Section 2.1 shows that the only functionally complete Boolean algebra is the two-element one. The functionally complete rings are the matrix rings over finite fields (Theorem 16 in Section 2.2), while the functionally complete groups are the finite simple non-Abelian ones (Theorem 18 in Section 2.3). Although these results were already known (see e.g. [40, 29, 26]), we introduce algorithmic proofs for them: we use these algorithms later in Chapter 3 to obtain upper bounds on lengths of polynomials. The last two Sections contain the results that there are no more functionally complete semigroups (Section 2.4) or semirings (Section 2.5) other than those already mentioned above for groups or rings:

Theorem. *(Theorem 28) Every finite functionally complete semigroup is a group.*

Theorem. *(Theorem 32) Every finite functionally complete semiring is a ring.*

In Chapter 3 we investigate the length of polynomials. We give upper and lower bounds on the lengths of shortest polynomials realizing special or arbitrary functions.

Definition. (Definition 35 and Definition 37) The *length* of a polynomial expression over $\mathbf{A}$ is defined recursively:

1. The length of a variable x or a constant c is 1: $\|x\|_{\mathbf{A}} = \|c\|_{\mathbf{A}} = 1$.
2. For an m-variable basic function f of $\mathbf{A}$ and for polynomial expressions $p_1, \ldots, p_m$, the length of $f(p_1, \ldots, p_m)$ is the sum of the lengths of p_i's: $\|f(p_1, \ldots, p_m)\|_{\mathbf{A}} = \sum_{i=1}^{m} \|p_i\|_{\mathbf{A}}$. Then the length of $f(x_1, \ldots, x_m)$ is $\|f\|_{\mathbf{A}} = m$.

The length of a function f over an algebra $\mathbf{A}$ is the length of a shortest polynomial p over $\mathbf{A}$ realizing the function f.

The two most important theorems which can be applied for functionally complete algebras in general are Theorem 45 and Theorem 48.

Theorem. *(part of Theorem 45) Let $\mathbf{A}$ be a functionally complete algebra and let 0 be an element of A. Let p be a shortest polynomial realizing an arbitrary n-ary function f over $\mathbf{A}$ with e-many non-zero values ($1 \leq e \leq |A|^n$). Then the following inequality holds:*

$$\|p\|_{\mathbf{A}} \leq c \cdot n^{c_1} \cdot e^{c_2},$$

where c, c_1 and c_2 are constants depending on the algebra $\mathbf{A}$ and on the element 0, $c_1 \geq 1$, $c_2 \geq 1$.

Theorem. *(Part of Theorem 48) Let $\mathbf{A}$ be a functionally complete ring or functionally complete Boolean algebra, $N = |A|$. Let p be a shortest polynomial realizing an arbitrary n-ary function f over $\mathbf{A}$ with e-many non-zero values, where $1 \leq e \leq N^n$. Then the following inequality holds:*

$$\|p\|_{\mathbf{A}} \leq e \cdot (1 + T \cdot (3 + n - \log_N e)) - 2 \cdot T,$$

where T is a constant depending on the algebra $\mathbf{A}$.

For an algebra $\mathbf{A}$ we denote by N the number of elements of A, i.e. $N = |A|$. Let 0 be an element of A. Theorem 45 bounds the length of an n-ary function by the product of some power of n, the number e of its non-zero values, and some constant depending on the algebra. In Theorem 48

we replace the factor of n^{c_1} by another factor: $(3+n-\log_N e)$. This new factor is linear in n, but it can be bounded by a constant unless e is really small compared to N^n. Therefore Theorem 48 gives a better upper bound; unfortunately it cannot be applied for arbitrary functionally complete algebras. Theorem 46 states that if $\mathbf{A}$ is a functionally complete algebra then for large enough n there exists an n-ary function which cannot be realized with a polynomial shorter than $c \cdot N^n \cdot (\log n)^{-1}$ for some constant c. Here and from now on we denote the base 2 logarithm function by log.

Then in the following Sections we derive bounds for every functionally complete algebra mentioned in Chapter 2. In Sections 3.2 and 3.3 by using Theorem 48 we obtain bounds on the length of arbitrary n-ary functions for the two-element Boolean algebra and for functionally complete rings.

Theorem. *(part of Theorem 61) Let* $\mathbf{B}$ *be the two-element Boolean algebra. Let* f *be an arbitrary* n*-ary function over* $\{0,1\}$ *with* e*-many non-zero values* $(1 \leq e \leq 2^n)$*. Then*

$$\|f\|_{\mathbf{B}} \quad \leq \quad (3+n-\log e) \cdot e - 2.$$

Theorem. *(part of Theorem 66) Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$ *and let* f *be an arbitrary* n*-ary function over* $\mathbf{F}$ *with* e*-many non-zero values. Then*

$$\|f\|_{\mathbf{F}} \leq 2 \cdot q \cdot \left(3+n-\log_q e\right) \cdot e$$

if $q \geq 3$ *and*

$$\|f\|_{\mathbf{F}} \leq 2 \cdot (3+n-\log e) \cdot e - 4$$

if $q = 2$*.*

Theorem. *(Theorem 68) Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$ *and let* $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$*, the* $k \times k$*-matrices over* $\mathbf{F}$ $(k \geq 2)$*. Let* $N = |\mathbf{M}_k(\mathbf{F})| = q^{k^2}$ *and let* f *be an arbitrary* n*-ary function over* $\mathbf{R}$ *with* e*-many non-zero values. Then*

$$\|f\|_{\mathbf{R}} \leq 16 \cdot (\log N)^{5/2} \cdot N^{1/4} \cdot (3+n-\log_N e) \cdot e.$$

Theorems 61, 66 and 68 have some common properties. Apart from the factor e and the strange factor $\left(n-\log_{|A|} e\right)$ there is only a constant factor, which is at most linear in the size of the particular algebra. On the other hand, in Section 3.4 the upper bound of Theorem 75 for groups is much worse compared to the case of rings or the two-element Boolean algebra.

Theorem. *(Part of Theorem 75) Let* $\mathbf{G}$ *be a functionally complete group. Let* $N = |\mathbf{G}|$*. Let* f *be an* n*-ary (possibly partial) function over* $\mathbf{G}$ *with*

e-many non-identity values ($1 \leq e \leq N^n$). Then the following inequalities hold:

$$\|f\|_{\mathbf{G}} \leq 2 \cdot K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}} \cdot V^2 \cdot (N-1)^{\log V} \cdot n^{\log V} \cdot e + 1,$$
$$\|f\|_{\mathbf{G}} \leq 6272 \cdot (K-1)^2 \cdot (N-1)^8 \cdot n^8 \cdot e + 1,$$

where $K_{G\setminus\{1\},b}$, $K_{b,G\setminus\{1\}}$ and V are constants depending on the group, $V \geq 4$ and $K = 1+\max\left\{ K_{G\setminus\{1\},b}, K_{b,G\setminus\{1\}} \right\}$ is bounded by the number of conjugacy classes of $\mathbf{G}$.

Apart from the factor e the bound contains a power of n and a constant, which is a power of the size of the group. This comparison of bounds seems to imply that groups are not the most efficient way of representing an arbitrary function; they seem to be less efficient than rings or the two-element Boolean algebra.

In Section 3.5 we investigate the special case when the finite simple non-Abelian group is an alternating group $\mathbf{A}_m$. We show in Section 3.1 that if a function can be realized by a polynomial over a group $\mathbf{G}_1$, and $\mathbf{G}_1 \leq \mathbf{G}_2$, then the same polynomial realizes the function over $\mathbf{G}_2$, too. This other realization has the same length, therefore when we try to find a shortest realization over a functionally complete group $\mathbf{G}$, we can as well just embed it into another functionally complete group and investigate realizations of the function over the larger group. Since every finite group can be embedded into $\mathbf{A}_m$ for some m, we dedicate a whole Section to investigate these groups.

Theorem. *(part of Theorem 88) Let $m \geq 5$ and let $N = |\mathbf{A}_m|$. Let f be an arbitrary (possibly partial) n-ary function over the group $\mathbf{A}_m$ with at most e-many non-identity values ($1 \leq e$). Then the following inequality holds:*

$$\|f\| \leq m \cdot \left(3N^2 - 9N + 8\right) \cdot \left(3n^2 - 3n + 2\right) \cdot e + 1.$$

If $4 \nmid m$, then we can replace the factor m by $\lfloor m/2 \rfloor$.

This bound is linear in e, but square in both n, N and m. This is the possible best bound we can obtain from Theorem 75, but still differs by a square factor of n and N from the case of rings or the two-element Boolean algebra.

We observe that the explanation for having worse bounds for groups can be derived from the main difference between rings and groups, namely that rings have two basic binary operations compared to only one for groups (which is closely related to the addition for rings). And it is indeed the case as Section 3.6 shows: in Theorem 101 we prove similar upper bounds on the length of an arbitrary function over a two-element set *if the commutator is considered as a basic operation.*

Theorem. *(Theorem 101) Let* $\mathbf{G} = (G, \cdot, ^{-1}, 1)$ *be a functionally complete group and let* $\mathbf{G}^c = (\mathbf{G}, [,]) = (G, \cdot, ^{-1}, 1, [,])$*, where* $[,]$ *is the commutator operation of* $\mathbf{G}$*. Let* $1 \neq u \in G$*, let* f *be an arbitrary* n*-ary function* $f\colon \{1, u\}^n \to \{1, u\}$ *with at most* e*-many non-identity values. Then*

$$\|f\|_{\mathbf{G}^c} \leq K_{G\setminus\{1\},u} \cdot ((10 + 3\,(n - \log e)) \cdot e - 5) + 1,$$

where $K_{G\setminus\{1\},u}$ *is a constant depending on the group* $\mathbf{G}$ *and on the element* u*. When* $\mathbf{G} = \mathbf{A}_m$ *(*$m \geq 5$*) and* u *is a 3-cycle, then*

$$\|f\|_{\mathbf{A}_m^c} \leq 4 \cdot ((10 + 3\,(n - \log e)) \cdot e - 5) + 1.$$

If $4 \nmid m$*, then we can replace the constant factor* 4 *by* 2*.*

The idea of Theorem 101 unfortunately cannot be used for an arbitrary function $f\colon G^n \to G$. We still can obtain better bounds than those in Theorem 75. The result looks similar to those in Theorem 45.

Theorem. *(Part of Theorem 103) Let* $\mathbf{G} = (G, \cdot, ^{-1}, 1)$ *be a functionally complete group and let* $\mathbf{G}^c = (\mathbf{G}, [,]) = (G, \cdot, ^{-1}, 1, [,])$*, where* $[,]$ *is the commutator operation of* $\mathbf{G}$*. Let* f *be an arbitrary* n*-ary (possibly partial) function over* $\mathbf{G}$ *with* e*-many non-identity values. Let* $N = |\mathbf{G}|$*. Then the following inequality holds:*

$$\|f\|_{\mathbf{G}^c} \leq 3 \cdot K^4 \cdot N \cdot n \cdot e,$$

where K *is a constant depending on the group* $\mathbf{G}$ *and is bounded by the number of conjugacy classes of* $\mathbf{G}$*. If* $\mathbf{G} = \mathbf{A}_m$ *(*$m \geq 5$*), then*

$$\|f\|_{\mathbf{A}_m^c} \leq 176 \cdot \lfloor m/2 \rfloor \cdot (N - 1) \cdot n \cdot e.$$

If $4 \nmid m$*, then we can replace the constant* 176 *by* 28*.*

These results not only show the importance of the commutator operation in groups, but they reveal that in some circumstances it behaves similarly as the multiplication behaves in a ring. Therefore a group with commutator can behave similarly to a ring. We use this property later on in the thesis.

The above results are relevant to the question of whether a computer based on a particular algebra offers a more efficient way of doing calculations than one, based on another type of algebra. *Efficiency*, however, can be defined in many ways. A natural way is to consider the length of polynomial expressions representing the desired function f. We are concerned

mostly about this aspect in Chapter 3. In Chapter 4 we investigate different computational models. In Section 4.1 we consider the *circuit complexity*.

For a functionally complete algebra $\mathbf{A}$ an $\mathbf{A}$-*circuit* C is a directed acyclic digraph with labelled nodes. The source nodes are labelled by variables or by constants, the other nodes (called 'gates') are labelled by basic operations of $\mathbf{A}$. A calculation at a gate is the application of the corresponding basic function on the values calculated at the sources of the incoming edges. Therefore a circuit computes a function at every gate. If every calculation at a gate takes one time-step, then the number $s(C)$ of gates (*size*) corresponds to the required time for calculating a function with a single processor machine. Similarly, the length $d(C)$ of a longest path (*depth*) corresponds to the time required to calculate a function with a multiple processor machine. For a function $f\colon A^n \to A^k$ let the *complexity of* f *with respect to* $\mathbf{A}$ be the size of a smallest n-ary $\mathbf{A}$-circuit which computes f; let the *depth of* f *with respect to* $\mathbf{A}$ be the depth of an n-ary $\mathbf{A}$-circuit which computes f and has the smallest depth. We denote the size of f by $s(f)$ and the depth of f by $d(f)$.

The main result of Section 4.1 is Theorem 117 which gives an upper bound on the size and the depth of an arbitrary n-ary function.

Theorem. *(part of Theorem 117) Let* $\mathbf{A}$ *be a functionally complete algebra,* $N = |A|$*. Let* $0 \in A$ *be an element. Let* f *be an arbitrary* n*-ary function over* $\mathbf{A}$ *with* e*-many non-zero values, where* $1 \leq e \leq |A|^n$*. Then the following inequalities hold:*

$$\begin{aligned} s(f) &\leq c_1 \cdot ((3 + n - \log_N e) \cdot e - 2), \\ s(f) &\leq c_2 \cdot n \cdot e, \\ d(f) &\leq c_3 \cdot \lceil \log e \rceil + c_4 \cdot \lceil \log n \rceil + c_5, \end{aligned}$$

where $c_1, \ldots, c_5$ *are constants depending on the algebra* $\mathbf{A}$ *and on the element* 0*.*

In Theorem 120 we give a lower bound on the size and the depth: we prove that for a functionally complete algebra $\mathbf{A}$ and for sufficiently large n there exist n-ary functions f_1 and f_2 such that $s_{\mathbf{A}}(f_1) \geq c \cdot N^n \cdot n^{-1}$ and $d_{\mathbf{A}}(f_2) \geq c' \cdot (n \log N - \log\log n)$ for some constants c and c'.

We refine our results for functionally complete groups in Section 4.2. Theorems 127 and 128 give sharper upper bounds on the depth and the size than Theorem 117.

Theorem. *(part of Theorems 127 and 128) Let* $\mathbf{G}$ *be a functionally complete group. Let* f *be an* n*-ary (possibly partial) function over* $\mathbf{G}$ *with* e*-many non-identity values. Let* $N = |\mathbf{G}|$ *and let* K *be the number of conjugacy classes*

in $\mathbf{G}$*. Then the following inequalities hold:*

$$s(f) \leq e \cdot (9nN \cdot (2K+7) - 7n - 7 + 4K) - 1,$$
$$d(f) \leq 14 + 2\log(K-1) + 8\log(N-1) + 8\log n + \log e.$$

Moreover if $\mathbf{G} = \mathbf{A}_m$ *(*$m \geq 5$*), then*

$$s(f) \leq e \cdot ((27N-14) \cdot n + m - 2) - 1,$$
$$d(f) \leq 1 + \log m + 2 \cdot (\log 3 + \log N + \log n) + \log e.$$

If $4 \nmid m$*, then we can replace the factor* $(27N-14)$ *by* $(13N-11)$ *and the factor* m *by* $2 \cdot \lfloor m/2 \rfloor$ *in the bound on the size.*

In Section 4.3 we compare the possible efficiency of functionally complete group based circuits and two-element algebra based circuits by simulating one with the other. Theorem 130 gives an upper bound on how much faster two-element algebra based circuits can be compared to circuit based on a functionally complete group.

Theorem. *(part of Theorem 130) Let* $\mathbf{G}$ *be a functionally complete group and let* K *be its number of conjugacy classes. Let* $\mathbf{A}$ *denote a two-element algebra whose basic operations are at most binary. Then there exists* $b \in G$*,* $b \neq 1$ *such that for every positive integer* n *and any function* $f\colon \{0,1\}^n \to \{0,1\}$ *we can find functions* p_1*,* p_2 *over* $\mathbf{G}$ *such that* p_1 *and* p_2 *are the same function over* $\{1,b\}$ *as* f *is over* $\{0,1\}$ *and*

$$s_{\mathbf{G}}(p_1) \leq (6K+456) \cdot s_{\mathbf{A}}(f), \qquad d_{\mathbf{G}}(p_2) \leq (14 + 2\log K) \cdot d_{\mathbf{A}}(f).$$

If $\mathbf{G} = \mathbf{A}_m$ *(for* $m \geq 5$*) and* $b = (1\,2\,3)$*, then for every positive integer number* n *and any function* $f\colon \{0,1\}^n \to \{0,1\}$ *we can find functions* p_1*,* p_2 *over* $\mathbf{G}$ *such that* p_1 *and* p_2 *are the same function over* $\{1,b\}$ *as* f *is over* $\{0,1\}$ *and*

$$s_{\mathbf{A}_m}(p_1) \leq 13 \cdot s_{\mathbf{A}}(f), \qquad d_{\mathbf{A}_m}(p_2) \leq 8 \cdot d_{\mathbf{A}}(f).$$

If $\mathbf{G} = \mathbf{A}_m$ *for* $m \geq 6$ *then we can choose* $b = (1\,2)(3\,4)$ *and we can replace the constants* 13 *and* 8 *by* 10 *and* 5*, respectively.*

Theorem 130 entails that, given that calculating basic operations take the same amount of time, computations based on the two-element Boolean algebra can be at most 13 times faster than computations based on the alternating group $\mathbf{A}_5$ and at most 10 times faster than computations based on the alternating group $\mathbf{A}_m$ (for $m \geq 6$). For the lower bound: Theorem 131

states that if the group multiplication of a functionally complete group $\mathbf{G}$ is computed by a circuit based on a two-element algebra, then the circuit has size at least $\lceil \log |\mathbf{G}| \rceil$.

In Section 4.4 we introduce a method by which a functionally complete group can simulate the ring $\mathbf{Z}_p$ for an odd prime p. For every ring-polynomial q we build an $\mathbf{A}_m$-circuit C (for $m \geq p+2$), which has linear size in $\|q\|$ and simulates the computation of the ring polynomial q. Whenever for some constant c we have $s_{\mathbf{Z}_p}(f) \leq c \cdot \|f\|_{\mathbf{Z}_p}$ or $d_{\mathbf{Z}_p}(f) \leq c \cdot \|f\|_{\mathbf{Z}_p}$, then we can compute f by an $\mathbf{A}_m$-circuit C, such that $s(C)$ is linear in $s_{\mathbf{Z}_p}(f)$ or $d(C)$ is linear in $d_{\mathbf{Z}_p}(f)$.

Theorem. *(Theorem 136) Let p be an odd prime and let $m \geq p+2$. Let $a = (1, \ldots, p) \in \mathbf{A}_m$, let r be a primitive root modulo p and let $h \in \mathbf{A}_m$ such that $a^h = a^r$. Let $\mathbf{H} = \langle h \rangle$ and let $\mathbf{A} = \langle a \rangle$. Let $in\colon \mathbf{Z}_p \hookrightarrow \mathbf{H} \times \mathbf{H}$ and $out\colon \mathbf{Z}_p \hookrightarrow \mathbf{A}$ be embeddings such that for every $0 \leq k \leq p-1$ we have $out(k) = a^k$ and $in(k) = \left(h^{k_1}, h^{k_2}\right)$ such that $r^{k_1} - r^{k_2} = k$ in $\mathbf{Z}_p$. Then for every $\mathbf{Z}_p$-polynomial $q(z_1, \ldots, z_n)$ there exists an $\mathbf{A}_m$-circuit C such that for every n-tuple $(r_1, \ldots, r_n)$ over $\mathbf{Z}_p$ the circuit C computes $out(q(r_1, \ldots, r_n))$ on the input $2n$-tuple $(in(r_1), \ldots, in(r_n))$ and*

$$s(C) \leq 16 \|q\|_{\mathbf{Z}_p},$$
$$d(C) \leq 8 \|q\|_{\mathbf{Z}_p}.$$

In Section 4.5 we investigate a different approach for function realizations than that introduced in Section 4.1. Krohn, Maurer and Rhodes in [22] showed a method how finite-state sequential circuits can be used for calculating an arbitrary Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. They, however, did not measure the efficiency of their method.

A *finite-state sequential circuit* is a 6-tuple $\mathbf{M} = (A, B, Q, q_0, \lambda, \mu)$, with basic input set A, basic output set B, state set Q, starting state q_0, next-state function $\lambda\colon Q \times A \to Q$ and output function $\mu\colon Q \to B$. Let A^+ be the free semigroup generated by A, i.e. all finite words with positive length constructed from the alphabet A. For any $t = a_1 \cdots a_n \in A^+$ let us define $\lambda'(t) : Q \to Q$ inductively: $\lambda'(a_1)(q) = \lambda(q, a_1)$ for $a_1 \in A$ and $q \in Q$. Let $\lambda'(a_1 \cdots a_k)(q) = \lambda'(a_k)(\lambda'(a_1 \cdots a_{k-1})(q))$ for $a_1 \ldots a_k \in A^+$ and $q \in Q$. Let $\mathbf{M}_q(a_1 \ldots a_k) = \mu(\lambda'(a_1 \ldots a_k)(q))$. This is the letter which machine $\mathbf{M}$ when started in state q outputs for the word $a_1 \ldots a_k$.

Let $\mathbf{F}(Q)$ denote the semigroup of all transformations of Q into itself under the multiplication $\cdot$, where for $f, g \in \mathbf{F}(Q)$ we have $(f \cdot g)(q) = g(f(q))$. Then $\lambda'\colon A^+ \to \mathbf{F}(Q)$ is a homomorphism: $\lambda'(a_1 \ldots a_k b_1 \ldots b_m) = \lambda'(a_1 \ldots a_k) \cdot \lambda'(b_1 \ldots b_m)$. Let us denote $\lambda'(A^+)$ by $\mathbf{M}^S$. We call $\mathbf{M}^S$ the semigroup of the machine $\mathbf{M}$.

Definition. (Definition 137) Let $\mathbf{M} = (A, B, Q, q_0, \lambda, \mu)$ be a finite-state sequential circuit. We say that $\mathbf{M}$ is a *simple non-Abelian Boolean circuit* if $A = B = \{0, 1\}$, $\mu(Q) = \{0, 1\}$, and $\mathbf{M}^S$ as a subsemigroup of $\mathbf{F}(Q)$ is a transitive simple non-Abelian group which is generated by two elements.

All simple non-Abelian Boolean circuits can be constructed in the following way: let $\mathbf{G}$ be a finite simple non-Abelian group generated by the elements g_0 and g_1. Let $\mathbf{H} \leq \mathbf{G}$ be a subgroup. Let us consider the right cosets of $\mathbf{H}$ in $\mathbf{G}$: let $R = \{\mathbf{H}g : g \in \mathbf{G}\}$. Let $\mu\colon R \to \{0, 1\}$ with $\mu(R) = \{0, 1\}$ be arbitrary. Then $\mathbf{M} = (\{0, 1\}, \{0, 1\}, R, \mathbf{H}, \lambda, \mu)$ is a simple non-Abelian Boolean circuit where $\lambda(\mathbf{H}g, k) = \mathbf{H}gg_k$ for $k = 0, 1$.

Definition. (Definition 139) Let $\mathbf{G}$ be a finite simple non-Abelian group, where the elements g_0 and g_1 generate $\mathbf{G}$. Let $\mathbf{M} = (\{0, 1\}, \{0, 1\}, R, \mathbf{H}, \lambda, \mu)$ be a simple non-Abelian Boolean circuit. Let p be an n-ary polynomial over $\mathbf{G}$ which does not contain inverses and every constant occurring in p is either g_0 or g_1. Then $B(\mathbf{M}, p) : \{0, 1\}^n \to \{0, 1\}$ is the Boolean function of n variables such that

$$B(\mathbf{M}, p)(y_1, \ldots, y_n) = \mathbf{M}_{\mathbf{H}}(p(g_{y_1}, \ldots, g_{y_n})) = \mu\big(\lambda'(p(g_{y_1}, \ldots, g_{y_n}))(\mathbf{H})\big).$$

In Theorem 140 we use the results of Chapter 3 for giving an upper bound on $\|p\|$.

Theorem. *(Theorem 140) Let $\mathbf{G}$ be a finite simple non-Abelian group, where the elements g_0 and g_1 generate $\mathbf{G}$. Let K be the number of conjugacy classes of $\mathbf{G}$ and let $N = |\mathbf{G}|$. Let $\mathbf{M} = (\{0, 1\}, \{0, 1\}, R, \mathbf{H}, \lambda, \mu)$ be a simple non-Abelian Boolean circuit such that $\mu(R) = \{0, 1\}$. Let $f\colon \{0, 1\}^n \to \{0, 1\}$ be an arbitrary function with e-many non-zero values. Then there exists a polynomial p over $\mathbf{G}$ such that p does not contain inverses and every constant in p is either g_0 or g_1, $f = B(\mathbf{M}, p)$, and*

$$\|p\| \leq 1\,605\,632 \cdot (N-1) \cdot (K-1)^2 \cdot n^8 \cdot e + (N-1).$$

If $\mathbf{G} = \mathbf{A}_m$ ($m \geq 5$), $\mathbf{H} = \mathbf{A}_{m-1}$, $g_0 = (1\,2\,3)$, and $g_1 = (3 \ldots m)$ (if $2 \nmid m$) or $g_1 = (1\,2)(3 \ldots m)$ (if $2 \mid m$) then we can choose p, such that

$$\|p\| \leq 128 \cdot \lfloor m/2 \rfloor \cdot n^2 \cdot e + (N-1).$$

1.2 The equivalence problem

Up to this point we were interested in finding polynomials which represent certain functions. Another interesting aspect is to find the functions represented by polynomials, more precisely to decide whether or not two given

polynomials define the same function. This problem is called the *polynomial equivalence problem* or *identity checking problem*. This question is interesting not only for functionally complete algebras, but for any algebra; and not only for polynomials, but for terms (expressions built up from variables and the basic operations of $\mathbf{A}$ using composition, but without constants) as well. This problem is called the *equivalence problem*. These questions are clearly decidable for any given finite algebra: one only has to check whether the two polynomials (or terms) attain the same value for every possible substitution from the given algebra. Thus the interesting question is whether or not this decision can be made in some fast way, i.e. to determine the computational complexity of deciding whether or not two polynomials (terms) represent the same function.

To every term or polynomial expression $t(x_1, \ldots, x_n)$ and each algebra $\mathbf{A}$ we denote the naturally associated function by $t^{\mathbf{A}}\colon A^n \to A$. We recall that an algebra $\mathbf{A}$ *satisfies* an equation $s(\vec{x}) \approx t(\vec{x})$ for $\vec{x} = (x_1, \ldots, x_n)$, if the corresponding functions $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are the same function. We denote this by $\mathbf{A} \models s \approx t$.

Definition. (Definition 141) *Equivalence problem* and *polynomial equivalence problem*.

> **Given:** A finite algebra $\mathbf{A}$.
>
> **Instance:** Two term expressions (for the equivalence problem), or two polynomial expressions (for the polynomial equivalence problem). Let the two expressions be s and t.
>
> **Question:** Do the two input expressions realize the same function over $\mathbf{A}$, i.e. does $\mathbf{A} \models s \approx t$ hold?

The complexity is always in coNP: for proving that two polynomials or terms are not realizing the same function it is enough to check a substitution where they differ. Similarly, it is easy to see that whenever the equivalence problem is coNP-complete, so is the polynomial equivalence problem. Moreover if the polynomial equivalence problem is in P, so is the equivalence problem.

In Section 5.3 we determine the complexity of the polynomial equivalence problem for functionally complete algebras.

Theorem. *(Theorem 146) The polynomial equivalence problem for a nontrivial functionally complete algebra* $\mathbf{A}$ *is coNP-complete.*

This is a joint result with Nehaniv and Szabó [14]. A corollary of this theorem is that the polynomial equivalence problem is coNP-complete for matrix rings over finite fields or for finite simple non-Abelian groups.

For finite commutative rings the computational complexity of the equivalence problem is completely characterized by Hunt and Stearnes [16]. They proved a dichotomy theorem: if a finite commutative ring is nilpotent, then the equivalence problem is in P; if it is not nilpotent, then it is coNP-complete. Later Burris and Lawrence generalized the result for arbitrary finite rings [2]. It follows from their proof that the same holds for the polynomial equivalence problem, too.

Much less is known for groups. There is a result of Burris and Lawrence [3] from 2004 that checking identities can be done in polynomial time for every finite nilpotent group and for the dihedral group $\mathbf{D}_n$ for odd n. It thus naturally arises to investigate the case of meta-Abelian groups. We carry out this examination in Chapter 6 and prove for several kinds of semidirect products that the complexity of the polynomial equivalence problem is in P. The following theorem summarizes the main results:

Theorem. *(Theorem 151 and Theorem 154) Let* $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ *such that the following hold:*

(a) $\mathbf{A}$ *is Abelian and either the exponent of* $\mathbf{A}$ *is squarefree or* $\mathbf{A}$ *is cyclic;*

(b) the polynomial equivalence problem for $\mathbf{B}$ *is in P;*

(c) for ever prime p *dividing the size of* $\mathbf{A}$ *and* $\mathbf{P} \in Syl_p(\mathbf{A})$ *the group* $\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})$ *is Abelian and* $p \nmid |\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})|$*, where* $C_{\mathbf{B}}(\mathbf{P})$ *denotes the centralizer of* $\mathbf{P}$ *in* $\mathbf{B}$*.*

Then the polynomial equivalence problem for $\mathbf{G}$ *is in P.*

Examples for such groups are the above-mentioned dihedral groups, the alternating group $\mathbf{A}_4$, or the wreath product of two cyclic groups. This is a joint result with Szabó [15].

These were results with polynomial time complexity. There are groups, for which the equivalence problem (and so the polynomial equivalence problem) is coNP-complete. In Chapter 7 we prove the following:

Theorem. *(Theorem 156) The equivalence problem for a finite nonsolvable group* $\mathbf{G}$ *is coNP-complete.*

From this result one wonders whether a dichotomy theorem, similar to the one for finite rings, holds for finite groups. At the moment this is an open question. Theorem 156 is a joint result with Lawrence, Mérai and Szabó [13].

In Section 3.6 we observed that the commutator as a basic operation can significantly change the length of realizing polynomials for several group-functions. For example, the expression $[[[x_1, x_2], x_3], \ldots, x_n]$ has length n if the commutator is a basic operation, but has exponential length in n when expressed by only the group multiplication. Such a decrease in the length suggests that the complexity of the equivalence problem might change if the commutator is a basic operation. Other group operations might have a similar property. A straightforward question arises, whether the complexity of the equivalence problem changes by taking one or more new operations as additional basic operations. Moreover, this question is interesting not only for groups but for all finite algebras. Hence we can raise the question in general:

Definition. (Part of Definition 166) Let $f_1, \ldots, f_n$ be polynomial expressions over the group $\mathbf{G}$. The algebra $(\mathbf{G}, f_1, \ldots, f_n)$ is defined to be the algebra $(G, \cdot, ^{-1}, 1, f_1, \ldots, f_n)$, i.e. the algebra with underlying set G and with basic operations $\cdot, ^{-1}, 1$ together with $f_1, \ldots, f_n$ as well.

1. The *extended equivalence problem* for $\mathbf{G}$.
   We say that the extended equivalence problem for $\mathbf{G}$ is in P if for all possible term expressions $f_1, \ldots, f_n$, built up from variables and the basic operations of $\mathbf{G}$, the equivalence problem over $(\mathbf{G}, f_1, \ldots, f_n)$ is in P (Theorem 168).

   We say that the extended equivalence problem for $\mathbf{G}$ is coNP-complete if there exist some term expressions $f_1, \ldots, f_n$, built up from variables and the basic operations of $\mathbf{G}$, such that the equivalence problem over $(\mathbf{G}, f_1, \ldots, f_n)$ is coNP-complete.

2. The *extended polynomial equivalence problem* for $\mathbf{G}$.
   We say that the extended polynomial equivalence problem for $\mathbf{G}$ is in P if for all polynomial expressions $f_1, \ldots, f_n$, built up from variables, constants from $\mathbf{G}$ and the basic operations of $\mathbf{G}$, the polynomial equivalence problem over $(\mathbf{G}, f_1, \ldots, f_n)$ is in P.

   We say that the extended polynomial equivalence problem for $\mathbf{G}$ is coNP-complete if there exist some polynomial expressions $f_1, \ldots, f_n$, built up from variables, constants from $\mathbf{G}$ and the basic operations of $\mathbf{G}$, such that the polynomial equivalence problem over $(\mathbf{G}, f_1, \ldots, f_n)$ is coNP-complete.

In Chapter 8 we consider the complexity of the extended equivalence problem and the extended polynomial equivalence problem for finite groups.

We start with nilpotent groups in Section 8.1. The (original) equivalence and the polynomial equivalence problems for finite nilpotent groups are in P by Burris and Lawrence [3]. Using the idea of their proof we prove that the extended polynomial equivalence problem is in P.

We prove in Chapter 7 that for non-solvable groups the equivalence problem is coNP-complete. As the extended problems are always at least as 'hard' as the original, we can conclude that the extended equivalence and the extended polynomial equivalence problems are coNP-complete for non-solvable groups. The complexity of the equivalence problem for non-nilpotent solvable groups is, for the most part, a *terra incognita* of mathematics. Only very few partial results are known (in Section 6.1 we proved that for a special class of meta-Abelian groups the complexity of the equivalence problem is in P, e.g. for meta-cyclic groups, dihedral groups $\mathbf{D}_{2k+1}$, $\mathbf{S}_3$ or $\mathbf{A}_4$), but we do not know the answer even for the symmetric group $\mathbf{S}_4$. The following theorem completes the characterization of the extended equivalence problem:

Theorem. *(Theorem 169) Let* $\mathbf{G}$ *be a finite solvable non-nilpotent group. Then there exists a term expression* f *(built up from variables and the basic operations of* $\mathbf{G}$*) such that the equivalence problem for* $(\mathbf{G}, f)$ *is coNP-complete.*

From these results we immediately have the following corollary:

Corollary. *(Corollary 170) Let* $\mathbf{G}$ *be a finite group. If* $\mathbf{G}$ *is nilpotent then the extended equivalence and the extended polynomial equivalence problems are in P. If* $\mathbf{G}$ *is not nilpotent then the extended equivalence and the extended polynomial equivalence problems are coNP-complete.*

The function f is not uniform in these proofs; it depends on the group $\mathbf{G}$. However, we show in Section 8.5 that for a large class of groups f can be chosen as the *commutator*.

Let us recall that the lower central series for a group $\mathbf{G}$ is the following sequence of normal subgroups: $\gamma_0(\mathbf{G}) = \mathbf{G}$, $\gamma_i(\mathbf{G}) = [\mathbf{G}, \gamma_{i-1}(\mathbf{G})]$. It is clear that if $i < j$, then $\gamma_i(\mathbf{G}) \geq \gamma_j(\mathbf{G})$. For every finite group the lower central series terminates in $\gamma_{i_0}(\mathbf{G})$ for some i_0. Let us denote this normal subgroup $\gamma_{i_0}(\mathbf{G})$ with $\mathbf{N} = \mathbf{N}(\mathbf{G})$.

Theorem. *(Theorem 184) Let* $\mathbf{G}$ *be a non-nilpotent group, let* $\mathbf{N} = \mathbf{N}(\mathbf{G})$ *be the final term of the lower central series as defined above. Let us suppose that* $\mathbf{N}$ *and* $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ *are both Abelian. Let us suppose that* $\exp(\mathbf{G}/F(\mathbf{G})) > 2$*, where* $F(\mathbf{G})$ *is the Fitting subgroup of the group* $\mathbf{G}$*. Then the equivalence problem for* $(\mathbf{G}, [,])$ *is coNP-complete, where* $[,]$ *denotes the commutator operation.*

Comparing the results of Section 8.5 to the results of Section 6.1 we can conclude that the complexity of the equivalence and the extended equivalence problems are not always the same. In Section 6.1 we prove that the equivalence problem for $\mathbf{A}_4$ is in P. By Theorem 184 the equivalence problem for $(\mathbf{A}_4, [,])$ is coNP-complete. Moreover we observe that if a corresponding theorem could be shown for any $\mathbf{G}$ such that $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ and $\mathbf{N}$ are both Abelian, and $\exp(\mathbf{G}/F(\mathbf{G})) = 2$, then Theorem 169 would follow by induction with f being the commutator of the group. If, however, it is not the case, then the characterization would be much harder.

1.3 The equation solvability problem

One of the oldest algebraic questions, equally important in computer science, is to decide whether or not an equation has a solution. This question again can be easily decided over finite algebras: one only has to check whether there is a substitution for which the two sides of the equation attain the same value. Thus the interesting question is again to determine the computational complexity of deciding whether two polynomials can attain the same value at some substitution. This is the *equation solvability problem* or *equation satisfiability problem.*

Definition. (Definition 142) *Equation solvability problem.*

Given: A finite algebra $\mathbf{A}$.

Instance: Two polynomial expressions p, q.

Question: Do the two input polynomials attain the same value for at least one substitution over $\mathbf{A}$, i.e. does the equation $p = q$ have a solution over $\mathbf{A}$?

The computational complexity of the equation solvability problem is NP-complete for functionally complete algebras. Nipkow asserted it in [29]; his proof, however, yields only a weaker theorem. In Section 5.2 we first give the theorem that follows from his proof, then give the complexity of the polynomial satisfiability problem for functionally complete algebras:

Theorem. *(Theorem 143) The equation solvability problem for a nontrivial functionally complete algebra* $\mathbf{A}$ *is coNP-complete.*

This is another joint result with Nehaniv and Szabó in the paper [14].

Surprisingly there are no published papers about the complexity of the equation solvability problem for finite rings. The complexity of the equation

solvability problem has been solved for nilpotent or non-solvable finite groups by Goldmann and Russell [10]. In Section 6.2 we determine the complexity of this problem for meta-Abelian groups with pq-many elements and prove the following:

Theorem. *(Theorem 155) For any group* $\mathbf{G}$ *of order* pq *where* p *and* q *are primes the equation solvability problem for* $\mathbf{G}$ *is in P.*

No other results are known about the complexity of the equation solvability problem for groups.

After comparing the complexity of the equivalence, polynomial equivalence, or equation solvability problems, one might think that if any of these three complexities is in P for a particular algebra, then the two other complexities are in P for the same algebra. This is, however, not the case: Seif and Szabó presented a 10 element semigroup (see [34]) for which the equivalence problem is in P and the equation solvability problem is NP-complete. Klíma proved an even stronger result in [20], where he showed a semigroup of size 24 for which the equation solvability problem is NP-complete but the polynomial equivalence problem is in P. An open question of the thesis is whether there exist such examples among groups.

1.4 Methods

Several methods appear throughout the thesis; many of them are used and recur for proving theorems from different areas of algebra or of computational complexity. We summarize them here and note how they are used.

Iterating functions in logarithmic depth. This is one of the most important methods used in Chapter 3. We observe that certain binary polynomial expressions can be iterated many times quite efficiently, i.e. in a way that the n-ary version of the polynomial expression will have polynomial length in n. Detailed description of the method can be found in Lemma 44. The method is used in the proof of Theorem 45, and in Section 3.4 for estimating the length of the '*and*' function over groups. The method is also used in Section 5.2, where we prove that the equation satisfiability problem is NP-complete over functionally complete algebras. Chapter 7 uses the method to prove that a certain n-ary commutator expression has polynomial length in n.

Recursive function realization. Theorem 48 gives better bounds than Theorem 45 as we realize the function recursively: we realize the n-ary function $f(x_1, \ldots, x_n)$ with the help of the $(n-1)$-ary functions

$f_a(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, a)$, where a is a constant element of the algebra. Detailed description of the method can be found in the proof of Theorem 48. The idea can be improved further in a way to obtain the theoretically best possible bounds. The method, however, cannot be used efficiently for every functionally complete algebra, it can only be applied for algebras with at least two binary basic operations. The method is useful for the two-element Boolean algebra or for rings, but not for groups, since groups have only one binary operation. This leads to the idea of taking the commutator as a basic operation, which is investigated in Section 3.6. We observe that using the commutator enables us to efficiently realize functions using recursion.

The commutator as a basic operation. In Section 3.6 we observe that taking the commutator as a basic operation can change the length of polynomial expressions significantly. The idea of the recursive function realization (as mentioned above) can be used efficiently, which shows that the group multiplication and commutator in some circumstances behave similarly as the ring addition and multiplication. In Section 8.5 we show that for many non-nilpotent groups the complexity of the equivalence problem is coNP-complete when using the commutator as a new basic operation. For many of these groups the complexity of the equivalence problem is in P. Therefore taking the commutator as a basic operation effectively changes the complexity, which is further evidence of the importance of the commutator in complexity questions on functions over groups.

Exploiting the endomorphism ring structure. In Chapter 6 we consider groups with structure $\mathbf{G} = \mathbf{A} \rtimes \mathbf{B}$, where $\mathbf{A}$ is Abelian. We reduce the equivalence problem of $\mathbf{G}$ to the equivalence problem over the endomorphism ring End $\mathbf{A}$. In Chapter 8 for any non-nilpotent group we find some Abelian subgroup $\mathbf{A} \leq \mathbf{G}$ and we polynomially reduce the extended equivalence problem over the original group to the equivalence problem over the endomorphism ring End $\mathbf{A}$. The third application of the method is used in Section 4.4, where we efficiently simulate the ring $\mathbf{Z}_p$ by an alternating group $\mathbf{A}_m$ for $m \geq p+2$. Again, we find an Abelian subgroup in $\mathbf{A}_m$ whose endomorphism ring is isomorphic to $\mathbf{Z}_p$.

Chapter 2

Functionally complete algebras

In this Chapter we give some general theorems about functionally complete algebras. Then we determine all functionally complete algebras for some classical structures, e.g. for Boolean algebras, rings, groups, semigroups and semirings. These theorems and proofs are used in Chapter 3, where we try to find short realizing polynomials for arbitrary functions.

Let $\mathbf{A}$ be a finite algebra with underlying set A (we usually denote the algebra by boldfaced capital letter and denote the underlying set by an italics capital letter). *Every algebra in the thesis is finite and contains at least two elements, unless we explicitly indicate otherwise.* Let p and q be two n-variable polynomial expressions over $\mathbf{A}$, i.e. expressions built up from variables, constants from $\mathbf{A}$ and the basic operations of $\mathbf{A}$ using composition. An equivalent definition is that an n-ary polynomial over $\mathbf{A}$ is a function built up the constant function, the projections and the basic operations of $\mathbf{A}$ using composition. The variable x_i corresponds to the ith projection $\pi_i \colon A^n \to A$, for which $\pi_i(x_1, \ldots, x_n) = x_i$. Both perspectives can be useful in different situations. By definition $\mathbf{A}$ is *functionally complete* if and only if every function over A can be expressed (or realized) as a polynomial of $\mathbf{A}$, i.e. for every nonnegative integer n and for every function $f \colon A^n \to A$ there is a polynomial expression $p(x_1, \ldots, x_n)$ over $\mathbf{A}$ such that for every n-tuple $(a_1, \ldots, a_n) \in A^n$ we have $p(a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$. We note that a nontrivial functionally complete algebra must contain an at least binary basic operation.

A term expression over an algebra $\mathbf{A}$ is an expression built up from variables (or projections) and the basic operations of $\mathbf{A}$. The difference between term expressions and polynomial expressions is that terms are not allowed to have constants, but polynomials are. If every possible function over A can be realized as a term expression of $\mathbf{A}$, then $\mathbf{A}$ is a *primal algebra*. Primality is a stronger assumption on an algebra than functional completeness, but

they coincide if $\mathbf{A}$ contains all constants as nullary basic operations. In the following we only consider functionally complete algebras.

Sometimes we add new functions over A to the algebra $\mathbf{A}$ as basic operations. If we add the functions $f_1, \ldots, f_n$ as new basic operations to the algebra $\mathbf{A}$, then we denote the algebra obtained by $(\mathbf{A}, f_1, \ldots, f_n)$.

Maurer and Rhodes proved in [26] that among nontrivial finite groups exactly the simple non-Abelian ones are functionally complete. They did not give a direct proof but proved a Stone–Weierstrass Theorem and as a corollary they obtained the functional completeness of the finite simple non-Abelian groups.

Definition 1. Let $\mathbf{A}$ be a finite algebra with underlying set A. Let S be an arbitrary set and let $\mathbf{F}(S, \mathbf{A})$ be the set of all functions $S \to A$. For every basic operation g of $\mathbf{A}$ we define g' over $\mathbf{F}(S, \mathbf{A})$ in the following way: if g is an n-ary operation and $f_1, \ldots, f_n \in \mathbf{F}(S, \mathbf{A})$, then $g'(f_1, \ldots, f_n) : S \to A$ and $g'(f_1, \ldots, f_n)(s) = g(f_1(s), \ldots, f_n(s))$. With these basic operations $\mathbf{F}(S, \mathbf{A})$ is an algebra with the same type as $\mathbf{A}$. We usually denote g' as g if it does not create confusion. We note that $\mathbf{F}(S, \mathbf{A})$ is isomorphic to the $|S|$-fold direct product $\mathbf{A}^S$ of $\mathbf{A}$ with itself.

Definition 2. Let $\mathbf{A}$ be a finite algebra and let S be a finite nonempty set. Let $\mathbf{F}$ be an arbitrary subalgebra of $\mathbf{F}(S, \mathbf{A})$, such that:

1. $\mathbf{F}$ contains the constant functions, namely for every $a \in \mathbf{A}$ there is a function $f_a \in \mathbf{F}$ such that for every $s \in S$ we have $f_a(s) = a$,

2. $\mathbf{F}$ separates every two elements of S, namely for every $s_1 \neq s_2 \in S$ there exists a function $f \in \mathbf{F}$ such that $f(s_1) \neq f(s_2)$.

If for every S these two properties imply that $\mathbf{F} = \mathbf{F}(S, \mathbf{A})$, then we say that $\mathbf{A}$ has the *Stone–Weierstrass property.*

In Section 2.3 we give a direct proof that the finite simple non-Abelian groups are the only functionally complete groups. Comparing it to the theorem in [26] we can conclude that functional completeness and the Stone–Weierstrass property are equivalent among finite groups. There are no direct proofs, whatsoever, for this equivalence in the literature; moreover the two properties are equivalent in general, namely

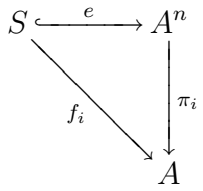
Theorem 3. *Let $\mathbf{A}$ be a finite algebra. Then $\mathbf{A}$ has the Stone–Weierstrass property if and only if $\mathbf{A}$ is functionally complete.*

Proof. Let us suppose that $\mathbf{A}$ has the Stone–Weierstrass property. Let $S = A^n$ for an arbitrary nonnegative integer n. Let $\mathbf{F}$ be a subalgebra of $\mathbf{F}(S, \mathbf{A}) = \mathbf{F}(A^n, \mathbf{A})$ which contains every constant function and every projection to a coordinate. Then $\mathbf{F}$ has both properties in Definition 2 and $\mathbf{A}$ has the Stone–Weierstrass property, hence $\mathbf{F} = \mathbf{F}(A^n, \mathbf{A})$. This is true for every nonnegative integer n, hence $\mathbf{A}$ is functionally complete.

Conversely, let us suppose now that $\mathbf{A}$ is functionally complete. Let S be a finite nonempty set and $\mathbf{F}$ be a subalgebra of $\mathbf{F}(S, \mathbf{A})$ which contains the constant functions and separates the elements of S. Let $n = \binom{|S|}{2}$ and let $f_{s_1,s_2} \in \mathbf{F}$ be a function for which $f_{s_1,s_2}(s_1) \neq f_{s_1,s_2}(s_2)$. Let $f_1, \ldots, f_n$ be an enumeration of these n-many functions. For every $a \in A$ let $f_a \colon S \to A$ be the constant a function: for every $s \in S$ let $f_a(s) = a$.

The idea is the following: we give an embedding $e \colon S \to A^k$ for some k. The embedding e will be defined in a way, such that f_i's become the composition of $\pi_i|_{\text{im } e}$ with e^{-1} (where $\pi_i \colon A^k \to A$ is the ith projection). Then using the functional completeness of $\mathbf{A}$, for an arbitrary function $f \colon S \to A$ we define a polynomial $p \colon A^k \to A$, built up by the projections and the constant functions, such that f is the composition of $p|_{\text{im } e}$ and e^{-1}. As p is built up from the projections and constant functions, so is f from f_i's and f_a's.

Let $e \colon S \to A^n$ be the following embedding of S to A^n: $e(s) = (f_1(s), \ldots, f_n(s))$. Note that if $\pi_i \colon A^n \to A$ is the projection to the ith coordinate, then $f_i = \pi_i \circ e$.



Now let $f \colon S \to A$ be an arbitrary function. We prove that $f \in \mathbf{F}$. Let $p \colon A^n \to A$ be a function such that $f = p \circ e$, i.e. for every $s \in S$ we have $f(s) = p(e(s))$. Such a function p exists, since e is an embedding. Now $\mathbf{A}$ is functionally complete, hence p is the composition of constant functions p_a, projections π_i and the basic functions of $\mathbf{A}$. Composing p with the embedding e we obtain f. In p replacing every π_i by f_i and every constant p_a by f_a yields that f is a composition of the functions $f_i = \pi_i \circ e$ $(1 \leq i \leq n)$ and $f_a = p_a \circ e$ $(a \in A)$. Since all f_i's and f_a's are in $\mathbf{F}$, $f \in \mathbf{F}$, too. □

The following proposition claims that a functionally complete algebra has no nontrivial homomorphism:

Proposition 4. *If a finite algebra* $\mathbf{A}$ *is functionally complete, then it has no nontrivial homomorphisms, namely if* $h\colon \mathbf{A} \twoheadrightarrow \mathbf{B}$ *is a surjective homomorphism, then either* h *is an isomorphism or* $|B| = 1$.

Proof. If a nontrivial homomorphism $h\colon \mathbf{A} \to \mathbf{B}$ exists, then there are 3 distinct elements $a_1, a_2, a_3 \in \mathbf{A}$ such that $h(a_1) = h(a_2) \neq h(a_3)$. Let $f\colon \mathbf{A} \to \mathbf{A}$ be any function such that $f(a_1) = a_1$, $f(a_2) = a_3$. Now, if f is represented by a polynomial p of $\mathbf{A}$, then by the interchangeability of p and h (which follows from the definition of homomorphism) we have

$$h\left(a_1\right) = h\left(p\left(a_1\right)\right) = p\left(h\left(a_1\right)\right) = p\left(h\left(a_2\right)\right) = h\left(p\left(a_2\right)\right) = h\left(a_3\right),$$

a contradiction. Hence if a nontrivial homomorphism h exists, then the above-mentioned f cannot be represented as a polynomial of $\mathbf{A}$ and $\mathbf{A}$ is not functionally complete. □

Remark 5. We note that an algebra has no nontrivial homomorphisms if and only if it is congruence-simple, i.e. it has no nontrivial congruence relations. Proposition 4 claims that every finite functionally complete algebra is congruence-simple. The converse holds for non-nilpotent rings (Section 2.2) and for non-Abelian groups (Section 2.3) but not in general, e.g. does not hold for semigroups (Section 2.4) or for semirings (Section 2.5).

The following theorem has been proved in [29] (first in [30]), but we discuss it, as we use the ideas of the proof later on.

Theorem 6. *Let* $\mathbf{A}$ *be an algebra, where* $|A| \geq 2$. *The algebra* $\mathbf{A}$ *is functionally complete if and only if the following three conditions hold:*

1. *there exist two distinct elements, called* 0 *and* 1,
2. *there exist two binomials (binary polynomials)* $+$ *and* $\cdot$ *such that* $0+a = a+0 = a$, $a \cdot 0 = 0$ *and* $a \cdot 1 = a$ *for every* $a \in A$,
3. *for every* $a \in A$ *there exists a monomial (unary polynomial)* χ_a *such that* $\chi_a(a) = 1$ *and* $\chi_a(b) = 0$ *if* $b \neq a$ *(the monomial characteristic functions).*

Proof. If $\mathbf{A}$ is functionally complete and $|A| \geq 2$, then assign 0 and 1 to two distinct elements of A and the polynomials described in the three conditions clearly exist.

If the three conditions hold, then we want to construct a polynomial for every n-variable function $f\colon A^n \to A$. First we prove that for every $a_1, \ldots a_n \in A$ there exists an n-variable polynomial $\chi_{a_1,\ldots,a_n}$ such that

$\chi_{a_1,\ldots,a_n}(a_1,\ldots,a_n) = 1$ and $\chi_{a_1,\ldots,a_n}(b_1,\ldots,b_n) = 0$ whenever $b_i \neq a_i$ for any $i \leq n$. Indeed,

$$\chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n) = \prod_{i=1}^{n} \chi_{a_i}(x_i) \tag{2.1}$$

has the property that if $x_i = a_i$ for every i then $\chi_{a_1,\ldots,a_n}$ is evaluated as 1, otherwise it is evaluated as 0.

We have to note though that since $\cdot$ is not necessarily associative or commutative, the meaning of $\prod$ is not straightforward. But it is easy to see that if we define $\prod$ as an iterated version of $\cdot$, then neither the ordering of the elements we multiply together nor the iteration method of the multiplication will change the fact that $\chi_{a_1,\ldots,a_n}$ defined with formula (2.1) will have the required property. Indeed, by the assumption we know that $1 \cdot 1 = 1$ and $0 \cdot 1 = 0$ by $a \cdot 1 = a$, moreover $1 \cdot 0 = 0$ and $0 \cdot 0 = 0$ by $a \cdot 0 = 0$. During the evaluation of $\chi_{a_1,\ldots,a_n}$ on some input we multiply 1's and 0's together. The result will be either 1 or 0, depending only on whether there were any 0's and not depending on the method or the ordering of the multiplication.

Now we create an n-variable polynomial p, which evaluates a given arbitrary n-variable function $f\colon A^n \to A$. Let p be the following:

$$p(x_1,\ldots,x_n) = \sum_{(a_1,\ldots,a_n)\in A^n} \left(f(a_1,\ldots,a_n) \cdot \chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n)\right),$$

where $\sum$ is an iterated version of $+$, the ordering of the elements or the iteration method is immaterial. It is clear that when we evaluate p on the input $(a_1,\ldots,a_n)$, then every summand will be 0 except one, which is $f(a_1,\ldots,a_n)$. So the sum is $f(a_1,\ldots,a_n)$, depending neither on the ordering of the summands nor on the method of the addition. □

Remark 7. We denote with $\chi_a(x)$ the characteristic function for which $\chi_a(a) = 1$ and $\chi_a(x) = 0$ if $x \neq a$. We denote the n-ary characteristic function with $\chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n)$, for which $\chi_{a_1,\ldots,a_n}(a_1,\ldots,a_n) = 1$ and $\chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n) = 0$ if $x_i \neq a_i$ for some i. These definitions however do not immediately make sense over groups, where we have an identity element corresponding to 0 (and the group multiplication naturally corresponds to the operation $+$ in Theorem 6), but no natural group element corresponds to 1. Hence for some $b \neq 1$ let $\chi_{a;b}(x)$ be the characteristic function for which $\chi_{a;b}(a) = b$ and $\chi_{a;b}(x) = 1$ if $x \neq a$. Let us denote the n-ary characteristic function with $\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n)$, for which $\chi_{a_1,\ldots,a_n;b}(a_1,\ldots,a_n) = b$ and $\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n) = 1$ if $x_i \neq a_i$ for some i. The semi-colon makes a difference between the two possible meanings of the indexes. We note here that there not necessarily exists a natural group operation which corresponds to the operation $\cdot$ in Theorem 6.

Corollary 8. *If* $\mathbf{R}$ *is a finite ring with an identity element, then* $\mathbf{R}$ *is functionally complete if and only if every* 1*-variable function can be expressed as a polynomial.*

Proof. If $\mathbf{R}$ is functionally complete, then every 1-variable function can be expressed as a polynomial. For the other direction we use Theorem 6: let us choose $+$ and $\cdot$ as the regular addition and multiplication of the ring, and let us choose 0 and 1 as the zero and unit-element of the ring. The monomial characteristic functions exist by the assumption that every 1-variable function can be expresses as a polynomial. □

There are other equivalent conditions for functional completeness. One of them is that the discriminator function can be expressed.

Theorem 9. *An algebra* $\mathbf{A}$ *is functionally complete if and only if there exists a three-variable polynomial* d *such that it is a discriminator, i.e.*

$$d(x,y,z) = \begin{cases} z, & \text{if } x = y \\ x, & \text{if } x \neq y \end{cases}.$$

Proof. It is clear that if $\mathbf{A}$ is functionally complete then the discriminator polynomial exists. Now assume that d is a discriminator polynomial, let 0 and 1 be two arbitrary (different) elements of A. We will use Theorem 6 and express $+$, $\cdot$ and χ_a using the discriminator d.

Let $x + y = d(y, 0, x)$. Now $x + 0 = d(0, 0, x) = x$ (hence $0 + 0 = 0$) and if $x \neq 0$ then $0 + x = d(x, 0, 0) = x$ as well.

Let $x \cdot y = d(0, d(0, y, 1), x)$. Now $x \cdot 0 = d(0, d(0, 0, 1), x) = d(0, 1, x) = 0$ and $x \cdot 1 = d(0, d(0, 1, 1), x) = d(0, 0, x) = x$.

Let $\chi_0(x) = d(0, x, 1)$ and $\chi_a(x) = d(0, d(a, x, 0), 1)$ for every $a \neq 0$. Now $\chi_0(0) = d(0, 0, 1) = 1$ and $\chi_0(b) = d(0, b, 1) = 0$ for every $b \neq 0$. Moreover (for every $a \neq 0$) we have $\chi_a(a) = d(0, d(a, a, 0), 1) = d(0, 0, 1) = 1$ and $\chi_a(b) = d(0, d(a, b, 0), 1) = d(0, a, 1) = 0$ for every $b \neq a$. □

Remark 10. We mention that if the polynomials $x + y$, $x \cdot y$ and $\chi_0(x)$ have the properties as in Theorem 6, and if $x - y$ can be expressed by a polynomial such that $x - y = 0$ if and only if $x = y$ and $1 - 0 = 1$, then χ_a and d can be expressed as follows:

$$\begin{aligned} \chi_a(x) &= \chi_0(x - a), \\ d(x,y,z) &= z \cdot \chi_0(x - y) + x \cdot (1 - \chi_0(x - y)). \end{aligned}$$

We summarize the conditions equivalent to functional completeness:

Theorem 11. *The following are equivalent.*

1. *The algebra* $\mathbf{A}$ *is functionally complete.*

2. *For every nonnegative integer* n *and for every function* $f\colon A^n \to A$ *there is a polynomial expression* $p(x_1, \ldots, x_n)$ *over* $\mathbf{A}$ *such that for every* n*-tuple* $(a_1, \ldots, a_n) \in A^n$ *we have* $f(a_1, \ldots, a_n) = p(a_1, \ldots, a_n)$.

3. *The algebra* $\mathbf{A}$ *has the Stone–Weierstrass property, i.e. for every finite nonempty set* S *if a subalgebra* $\mathbf{F}$ *of* $\mathbf{A}^S$ *has the following two properties then* $\mathbf{F} = \mathbf{A}^S$*:*

   (a) $\mathbf{F}$ *contains the constant functions, namely for every* $a \in \mathbf{A}$ *there is a function* $f_a \in \mathbf{F}$ *such that for every* $s \in S$ *we have* $f_a(s) = a$.

   (b) $\mathbf{F}$ *separates every two elements of* S*, namely for every* $s_1 \neq s_2 \in S$ *there exists a function* $f \in \mathbf{F}$ *such that* $f(s_1) \neq f(s_2)$.

4. *The following three conditions hold:*

   (a) *there exist two distinct elements, called* 0 *and* 1,

   (b) *there exist two binomials (binary polynomials)* $+$ *and* $\cdot$ *such that* $0 + a = a + 0 = a$, $a \cdot 0 = 0$ *and* $a \cdot 1 = a$ *for every* $a \in A$,

   (c) *for every* $a \in A$ *there exists a monomial (unary polynomial)* χ_a *such that* $\chi_a(a) = 1$ *and* $\chi_a(b) = 0$ *if* $b \neq a$ *(the monomial characteristic functions).*

5. *The three-variable discriminator polynomial exists:*

$$d(x, y, z) = \begin{cases} z, & \text{if } x = y \\ x, & \text{if } x \neq y \end{cases} .$$

In the following Sections of this Chapter we determine all functionally complete algebras for different classes. We start with one of the most well-known class, the Boolean algebras.

2.1 Boolean algebras

It is well-known that the two-element Boolean algebra is functionally complete. This is the main reason that we can build universal machines based on this algebra. In this section we determine all finite functionally complete Boolean algebras. Our main reference on Boolean algebras and their representation is [36].

Definition 12. A Boolean algebra is a distributive complemented lattice, i.e. it has two binary operations $\wedge$ (*meet* or *and*), $\vee$ (*join* or *or*), a unary operation $\neg$ (*complement* or *not*) and two distinct elements 0 (or *false*) and 1 (or *true*), such that $\wedge$ and $\vee$ are both associative, commutative, moreover they satisfy both distributive laws, the absorption laws ($a \wedge (a \vee b) = a$ and $a \vee (a \wedge b) = a$), $a \wedge \neg a = 0$ and $a \vee \neg a = 1$.

Remark 13. The reader might wonder why we consider the quite specific Boolean algebras instead of e.g. the more general concept of lattices. It is easy to see, that there are no functionally complete lattices, as the lattice operations are order-preserving considering the usual partial ordering on the lattice. Hence, to make lattices as candidates for functionally complete algebras, we have to include at least one more operation which does not preserve the lattice partial ordering. The complement fulfills this requirement. We note that a complemented lattice does not have to be distributive, such as Boolean algebras are. Nevertheless, the most common complemented lattices are Boolean algebras; we only consider them in this Section.

We denote the two-element Boolean algebra by $\mathbf{B} = (\{0, 1\}, \wedge, \vee, \neg)$. By Stone's Representation Theorem we know that every finite Boolean algebra has of order 2^k for some positive integer k, and it is isomorphic with the complemented lattice of all subsets of the set $\{1, \ldots, k\}$. Moreover, the 2^k-element Boolean algebra is isomorphic with $\mathbf{B}^k$. The following theorem states that the only functionally complete Boolean algebra is the two-element algebra $\mathbf{B}$.

Theorem 14. *If* $\mathbf{A}$ *is a finite Boolean algebra, then* $\mathbf{A}$ *is functionally complete if and only if* $\mathbf{A} = \mathbf{B}$.

Proof. The 'only if' part is quite easy, as by Stone's Representation Theorem we know that $\mathbf{A} = \mathbf{B}^k$ for some positive integer k. Since the projections are nontrivial homomorphisms, applying Proposition 4 we have that $k = 1$, so the only possible candidate is $\mathbf{A} = \mathbf{B}$.

For the other direction we use Theorem 6. Let $x + y = (x \wedge \neg y) \vee (\neg x \wedge y)$ and let $x \cdot y = x \wedge y$. These are the usual mod 2 addition and multiplication operations over the set $\{0, 1\}$ and satisfy the conditions in Theorem 6. Moreover the two unary characteristic functions can be expressed as $\chi_0(x) = \neg x$ and $\chi_1(x) = x$. By Theorem 6 we have that $\mathbf{B}$ is functionally complete. □

Remark 15. We note that if we consider $\mathbf{A} = \mathbf{B}^k$ and we take the ith projection $\pi_i \colon \mathbf{A} \to \mathbf{A}$, $\pi_i(a_1, \ldots, a_n) = (0, \ldots, 0, a_i, 0, \ldots, 0)$ for every $1 \leq i \leq k$

as a basic operation, then the obtained algebra $(\mathbf{A}, \pi_1, \ldots, \pi_k)$ is functionally complete. Expressing any function $f\colon \left(B^k\right)^n \to B^k$ over $(\mathbf{A}, \pi_1, \ldots, \pi_k)$ is, however, not essentially different than expressing every coordinate of f over $\mathbf{B}$ and taking the $\vee$ of their projections, hence we do not pay any more attention to these algebras.

Second proof of Theorem 14. We give the discriminator operation:

$$d(x,y,z) = (((x \wedge \neg y) \vee (\neg x \wedge y)) \wedge x) \vee ((x \vee \neg y) \wedge (\neg x \vee y) \wedge z)\,.$$

It is easy to see that $d(x,y,z) = z$ if $x = y$ and $d(x,y,z) = x$ if $x \neq y$. □

$\mathbf{B}$ is not the only functionally complete algebra with 2-elements. If an algebra over $\{0,1\}$ can express the basic operations of $\mathbf{B}$, then it is functionally complete, too. In Section 2.2 we prove that the two-element field is another such example. The multiplication of this field is the same as $\wedge$ and the addition is the same as in the proof $(x + y = (x \wedge \neg y) \vee (\neg x \wedge y))$. It is sometimes called *xor* (*exclusive or*), too.

We consider here another two-element functionally complete algebra, as it has the most important practical application in Computer Science. Consider the algebra $\mathbf{B}_0 = (\{0,1\}, \mathrm{NAND}, \mathrm{NOR})$, where $x\,\mathrm{NAND}\,y = \neg(x \wedge y)$ (*negation of and*) and $x\,\mathrm{NOR}\,y = \neg(x \vee y)$ (*negation of or*). This algebra is functionally complete: either only NAND or only NOR is already enough to express $\wedge$, $\vee$ and $\neg$, as the following equations show:

$$\neg x = x\,\mathrm{NAND}\,1 = x\,\mathrm{NOR}\,0 \tag{2.2}$$

$$x \wedge y = (x\,\mathrm{NAND}\,y)\,\mathrm{NAND}\,1 = (x\,\mathrm{NOR}\,0)\,\mathrm{NOR}\,(y\,\mathrm{NOR}\,0) \tag{2.3}$$

$$x \vee y = (x\,\mathrm{NAND}\,1)\,\mathrm{NAND}\,(y\,\mathrm{NAND}\,0) = (x\,\mathrm{NOR}\,y)\,\mathrm{NOR}\,0 \tag{2.4}$$

These equations show that not only $\mathbf{B}_0$, but $\mathbf{B}_{\mathrm{NAND}} = (\{0,1\}, \mathrm{NAND})$ and $\mathbf{B}_{\mathrm{NOR}} = (\{0,1\}, \mathrm{NOR})$ are functionally complete, too. Today's computers are based on $\mathbf{B}_0$ as the NAND and NOR operations can be realized quite easily in practice [12]. In later Chapters we mainly consider $\mathbf{B}$ and $\mathbf{B}_0$.

2.2 Rings

In this Section we determine the functionally complete rings. We note that we do not require that the ring has an identity, the proofs work without it. The only notion we use, which is usually not considered for rings without an identity, is the Jacobson radical. The Jacobson radical can be defined for rings without identity, the same properties (which make sense without identity) can be proved and the Wedderburn–Artin Theorem holds, too [17].

The following theorem has a similar proof in [29], but we give a proof here, as it is quite algorithmic and gives an explicit way of realizing an arbitrary function over a functionally complete ring.

Theorem 16. *A finite ring* $\mathbf{R}$ *is functionally complete if and only if* $\mathbf{R}$ *is a matrix ring over a finite field.*

Proof. Suppose that the finite ring $\mathbf{R}$ is functionally complete. First we prove that $\mathbf{R}$ has no nontrivial two-sided ideals. Indeed, suppose that $\mathbf{I} \lhd \mathbf{R}$ and fix two elements (a, b) of $\mathbf{R}$ such that $0 \neq a \in \mathbf{I}$, $b \notin \mathbf{I}$. Let f be a function over $\mathbf{R}$ with the property that $f(0) = 0$, $f(a) = b$. Now if f can be represented with a polynomial p over $\mathbf{R}$, then consider $p + \mathbf{I}$ over $\mathbf{R}/\mathbf{I}$. Now $b + \mathbf{I} = p(a) + \mathbf{I} = p(a + \mathbf{I}) = p(\mathbf{I}) = p(0) + \mathbf{I} = \mathbf{I}$, which is a contradiction, since $b \notin \mathbf{I}$. This follows from Proposition 4, too.

Let $\mathbf{J}$ be the Jacobson radical of $\mathbf{R}$. Since $\mathbf{J}$ is a two-sided ideal of $\mathbf{R}$, $\mathbf{J}$ is either $\mathbf{R}$ or $\{\, 0 \,\}$. If $\mathbf{J} = \mathbf{R}$, then $\mathbf{R}$ is nilpotent, i.e. there exists some positive integer d such that the term $x_1 x_2 \dots x_d$ is evaluated as 0 whenever the variables attain values from $\mathbf{R}$. In this case, we can give an upper bound to the number of polynomials with n variables: Let N be the number of elements of $\mathbf{R}$, then there are at most $(N + n)^k$ monomials with length k (under monomial with length k we mean a product of k members, each member is either an element of the ring or a variable), hence there are $(N + n) + (N+n)^2 + \dots + (N+n)^{d-1} < (N+n)^d$ monomials which contain variables from the set $\{\, x_1, \dots, x_n \,\}$ and have length less than d. Every polynomial with at most n variables can be written as a sum of these monomials. Adding up the monomial m k-many times (where k is an integer) can be written as $k' \cdot m$, where k' is k modulo N (observe that $N \cdot r = 0$ in $\mathbf{R}$). Using this form every monomial has a non-negative integer coefficient between 0 and $N - 1$. This means that there exist at most $N^{(N+n)^d}$ polynomials over $\mathbf{R}$. On the other hand, it is easy to see that there exist N^{N^n}-many $R^n \to R$ functions. Since $N^{(N+n)^d} < N^{N^n}$ for large enough n, $\mathbf{R}$ is not functionally complete if $\mathbf{J} = \mathbf{R}$. Hence $\mathbf{J} = \{\, 0 \,\}$.

By the Wedderburn–Artin Theorem [17] we have that $\mathbf{R}/\mathbf{J} = \mathbf{R}$ is a direct sum of matrix rings over finite fields. Since any summand of this representation is a two-sided ideal, we can conclude that if $\mathbf{R}$ is functionally complete then it is a finite matrix ring $\mathbf{M}_k(\mathbf{F})$ over a finite field $\mathbf{F}$.

For the other direction let $\mathbf{R}$ be a finite matrix ring $\mathbf{M}_k(\mathbf{F})$, where $q = |\mathbf{F}|$. By Theorem 6 we only need to check if there exist polynomials $\chi_M(X)$ with the property that $\chi_M(M) = 1$ and $\chi_M(N) = 0$ if $N \neq M$, where N and M are $k \times k$ matrices over the finite field $\mathbf{F}$. Let us denote the identity matrix with I and let $I_{i,j}$ denote the matrix whose only non-zero value is 1 and is

in row i and column j. Let us denote with $M(i,j)$ the element of a matrix M which lies in row i and column j. Now with the following polynomial we can check the element $M(i,j)$ of a matrix M:

$$p_{i,j}(X) = \sum_{s=1}^{k} I_{s,i} \cdot X \cdot I_{j,s}.$$

It is easy to see that for any $k \times k$ matrix M we have $p_{i,j}(M) = M(i,j) \cdot I$. Now let $X \vee Y = X + Y - X \cdot Y$, and let $\bigvee_{i=1}^{n} X_i$ be the iterated version of $\vee$, the ordering or iteration method can be arbitrary. Observe that $\vee$ acts like the *or* function if we substitute only I and 0 (i.e. $I \vee 0 = 0 \vee I = I \vee I = I$ and $0 \vee 0 = 0$). Using the fact that u^{q-1} is either 1 (if $u \neq 0$) or 0 (if $u = 0$) we are able to check whether a matrix is 0 or not:

$$\delta(X) = \bigvee_{i,j=1}^{k} \left(p_{i,j}(X)^{q-1}\right) = \begin{cases} 0, & \text{if } X = 0 \\ I, & \text{if } X \neq 0 \end{cases}.$$

Finally we can find a realizing polynomial for χ_M:

$$\chi_M(X) = I - \delta(X - M) = \begin{cases} I, & \text{if } X = M \\ 0, & \text{if } X \neq M \end{cases}.$$

Whenever $\mathbf{R} = \mathbf{F}$ is a finite field containing q elements, then $\chi_a(x)$ has a quite simple representing polynomial:

$$\chi_a(x) = 1 - (x - a)^{q-1}.$$

□

Remark 17. We note here that the proof actually shows that for finite non-nilpotent rings the congruence-simple property is equivalent with the functional completeness. The two properties are not equivalent for finite rings, as e.g. 0-multiplication rings of prime order are congruence-simple but not functionally complete.

Second proof of Theorem 16. We give the discriminator operation (using the notations of the previous proof of Theorem 16). If $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$ a matrix ring over a finite field $\mathbf{F}$:

$$d(X,Y,Z) = \delta(X - Y) \cdot X + (I - \delta(X - Y)) \cdot Z.$$

If $\mathbf{R} = \mathbf{F}$ a finite field containing q elements, then expressing d is even more simple

$$d(x,y,z) = (x - y)^{q-1} \cdot x + \left(1 - (x - y)^{q-1}\right) \cdot z.$$

□

2.3 Groups

The following theorem gives us the functionally complete groups. We do not repeat the first proof of [26], but show another one (based on Exercise 14 on page 158 of [27]), which gives us an algorithm for finding realizing polynomials for an arbitrary function.

Theorem 18. *A finite group* $\mathbf{G}$ *is functionally complete if and only if* $\mathbf{G}$ *is simple and non-Abelian.*

Proof. Suppose that $\mathbf{G}$ is not simple, i.e. $\mathbf{N}$ is a nontrivial normal subgroup in $\mathbf{G}$. Fix $1 \neq a \in \mathbf{N}$ and $b \notin \mathbf{N}$. Let f be a unary function such that $f(x) = 1$ if $x \neq a$, and $f(a) = b$. If f can be represented with a polynomial p over the group $\mathbf{G}$ then consider $p/\mathbf{N}$ over $\mathbf{G}/\mathbf{N}$. Now $b\mathbf{N} = p(a)\mathbf{N} = p(a\mathbf{N}) = p(\mathbf{N}) = p(1)\mathbf{N} = \mathbf{N}$, which is a contradiction, since $b \notin \mathbf{N}$. This follows from Proposition 4, too.

Suppose that $\mathbf{G}$ is Abelian and let $1 \neq a \in \mathbf{G}$. Now if a function $f(x, y)$ has the property that $f(1, 1) = f(1, a) = f(a, 1) = 1$, $f(a, a) = a$, then it cannot be represented with a polynomial over $\mathbf{G}$: every two-variable polynomial has a form of $p(x, y) = x^{k_1} \cdot y^{k_2} \cdot c$. Now if $p(x, y) = x^{k_1} \cdot y^{k_2} \cdot c$ and $p(1, 1) = p(1, a) = p(a, 1) = 1$, then $c = 1^{k_1} \cdot 1^{k_2} \cdot c = p(1, 1) = 1$, $a^{k_1} = a^{k_1} \cdot 1^{k_2} \cdot c = p(a, 1) = 1$, $a^{k_2} = 1^{k_1} \cdot a^{k_2} \cdot c = p(1, a) = 1$, hence $p(a, a) = a^{k_1} \cdot a^{k_2} \cdot c = 1$.

Now suppose that $\mathbf{G}$ is a simple, non-Abelian group. We will prove the theorem via the following lemmas:

Lemma 19. *For every* $1 \neq u \in \mathbf{G}$ *and* $v \in \mathbf{G}$ *there are* $y_1, \ldots, y_k$ *such that* $v = u^{y_1} \cdots u^{y_k}$.

Proof. Let C_u be the conjugacy class of u in $\mathbf{G}$, and $\mathbf{H}_u$ the subgroup generated by C_u. If $u \neq 1$ then $\mathbf{H}_u \neq 1$. Now $\mathbf{H}_u$ is closed under conjugation, because its generator set is closed, too. Thus $\mathbf{H}_u \lhd \mathbf{G}$, and $\mathbf{G}$ is simple, hence $\mathbf{H} = \mathbf{G}$, which is equivalent with the statement of the lemma. □

Let $p_{u,v}(x) = x^{y_1} \cdots x^{y_k}$. Now we have $p_{u,v}(1) = 1$, $p_{u,v}(u) = v$.

Lemma 20. *For every* $u \neq 1 \neq v$ *in* $\mathbf{G}$ *there exists* $y \in \mathbf{G}$ *such that* $[u, v^y] \neq 1$.

Proof. $[u, v^y] = 1$ for every y means that u centralizes C_v, thus $u \in C_{\mathbf{G}}(C_v)$. For every subset $X \subseteq \mathbf{G}$ the centralizer of X is the same as the centralizer of $\langle X \rangle$. Now $v \neq 1$ and $C_{\mathbf{G}}(C_v) = C_{\mathbf{G}}(\langle C_v \rangle) = C_{\mathbf{G}}(\mathbf{G}) = Z(\mathbf{G})$ is the center of $\mathbf{G}$. Since $Z(\mathbf{G}) = 1$, $u \in C_{\mathbf{G}}(C_v)$ implies $u = 1$, a contradiction. □

Lemma 21. *For every $1 \neq b \in \mathbf{G}$ and for every natural number n there exists a polynomial $f_b^{(n)}(x_1, \ldots, x_n)$ such that $f_b^{(n)}(y_1, \ldots, y_n) = 1$, whenever $y_i = 1$ for some i, and $f_b^{(n)}(b, \ldots, b) = b$.*

Proof. Let $u_1 = b$, we define u_i for $i \leq n$ inductively such that $u_i \neq 1$ for every i. By Lemma 20 there exists c_i such that $[u_{i-1}, b^{c_i}] \neq 1$. Choose c_i and let $u_i = [u_{i-1}, b^{c_i}] \neq 1$. Let $h_1(x_1) = x_1$ and for every $2 \leq k \leq n$ let $h_k(x_1, \ldots, x_k) = [h_{k-1}(x_1, \ldots, x_{k-1}), x_k^{c_k}]$. With these notations we have that $h_k(b, \ldots, b) = u_k$, and if we substitute $x_i = 1$, then for every $k \geq i$ we have $h_k(x_1, \ldots, x_k) = 1$. By Lemma 19 we have a unary polynomial $p_{u_n,b}$ such that $p_{u_n,b}(1) = 1$, $p_{u_n,b}(u_n) = b$. With this notation $f_b^{(n)}(x_1, \ldots, x_n) = p_{u_n,b}(h_n(x_1, \ldots, x_n))$ satisfies the conditions of the lemma. □

Remark 22. It is easy to see that for any $b \in \mathbf{G}$ the function $f_b^{(2)}$ described above is the '*and*' function if we encode '*false*' with 1 and '*true*' with b. This is in fact a function we cannot obtain as a polynomial expression if $\mathbf{G}$ is Abelian.

Lemma 23. *For every $1 \neq b \in \mathbf{G}$ there exists a unary polynomial $\chi_{1;b}$ such that $\chi_{1;b}(1) = b$ and $\chi_{1;b}(u) = 1$ for all $1 \neq u \in \mathbf{G}$.*

Proof. Let $\mathbf{G} = \{u_1, \ldots, u_N\}$, where $u_1 = 1$. By Lemma 19 we have the unary polynomials $p_{u_i,b}$ such that $p_{u_i,b}(1) = 1$ and $p_{u_i,b}(u_i) = b$. By Lemma 21 we have the $N-1$-ary polynomial $f_b^{(N-1)}$ such that $f_b^{(N-1)}(b, \ldots, b) = b$ and $f_b^{(N-1)}(y_1, \ldots, y_n) = 1$, whenever for some i, $y_i = 1$. Take $\chi_{1;b}(x) = f_b^{(N-1)}\left(bp_{u_2,b}(x)^{-1}, \ldots, bp_{u_N,b}(x)^{-1}\right)$. Now if we substitute $x = 1$ then for every i we have $bp_{u_i,b}(x)^{-1} = b$, hence $\chi_{1;b}(1) = f_b^{(N-1)}(b, \ldots, b) = b$. If we substitute $x = u_i$, then $bp_{u_i,b}(x)^{-1} = bb^{-1} = 1$, hence $\chi_{1,b}(u_i) = f_b^{(N-1)}\left(bp_{u_2,b}(u_i)^{-1}, \ldots, 1, \ldots, bp_{u_N,b}(u_i)^{-1}\right) = 1$. □

Remark 24. $\chi_{1;b}$ is clearly the 1-variable characteristic function for the identity element. Among rings it is quite clear that a characteristic function attains values 0 or 1. However among groups it is not the case. From now on we denote with $\chi_{a_1,\ldots,a_n;b}(x_1, \ldots, x_n)$ the characteristic function for which $\chi_{a_1,\ldots,a_n;b}(a_1, \ldots, a_n) = b$ and $\chi_{a_1,\ldots,a_n;b}(x_1, \ldots, x_n) = 1$ whenever $x_i \neq a_i$ for some i. The semi-colon makes a difference between the two possible meanings of the indexes.

Lemma 25. *For every $a_1, \ldots, a_n; b \in \mathbf{G}$ there exists an n-ary polynomial $\chi_{a_1,\ldots,a_n;b}$ such that $\chi_{a_1,\ldots,a_n;b}(a_1, \ldots, a_n) = b$ and $\chi_{a_1,\ldots,a_n;b}(x_1, \ldots, x_n) = 1$ whenever $x_i \neq a_i$ for some i.*

Proof. Let us fix some $b \neq 1$. By Lemma 21 we have the n-ary polynomial $f_b^{(n)}$ such that $f_b^{(n)}(b,\ldots,b) = b$ and $f_b^{(n)}(y_1,\ldots,y_n) = 1$, whenever for some i, $y_i = 1$. By Lemma 23 we have the unary polynomial $\chi_{1;b}$ such that $\chi_{1;b}(1) = b$ and $\chi_{1;b}(x) = 1$, whenever $x \neq 1$. Let $\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n) = f_b^{(n)}\left(\chi_{1;b}\left(x_1 a_1^{-1}\right),\ldots,\chi_{1;b}\left(x_n a_n^{-1}\right)\right)$. Now if we substitute $x_i = a_i$ for every i, then $\chi_{a_1,\ldots,a_n;b}(a_1,\ldots,a_n) = f_b^{(n)}(\chi_{1;b}(1),\ldots,\chi_{1;b}(1)) = f_b^{(n)}(b,\ldots,b) = b$. If for some i we substitute $x_i \neq a_i$, then we have $\chi_{1;b}\left(x_i a_i^{-1}\right) = 1$ thus $\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n) = 1$, as requested. □

Now let $f\colon G^n \to G$ be an arbitrary n-ary function. Then

$$p(x_1\ldots,x_n) = \prod_{\substack{(a_1,\ldots,a_n)\in G^n \\ 1\neq u=f(a_1\ldots,a_n)}} \chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n)$$

is a representing polynomial for f as

$$p(a_1,\ldots,a_n) = \chi_{a_1,\ldots,a_n;f(a_1,\ldots,a_n)}(a_1,\ldots,a_n) = f(a_1,\ldots,a_n).$$

□

Remark 26. We note here that the proof actually shows that for finite non-Abelian groups the congruence-simple property is equivalent with the functional completeness. These two properties are not equivalent for all finite groups as e.g. finite groups with prime order are congruence-simple but not functionally complete.

Second proof of Theorem 18. We will use Theorem 6 and express $+$, $\cdot$ and χ_a for every group element a.

Let 1 be the zero-element, let us fix a b element of a group as the unit-element and let $+$ be the multiplication of the group. Now, $\chi_{1,b}(a^{-1}x)$ will do as the characteristic function for a. All we need now is the $\cdot$ function with the following properties: $\cdot(x,1) = 1$, $\cdot(x,b) = x$. For every $a \neq 1$ let c be an element of the group such that $[b,b^c] \neq 1$. Let f_a be the following function: $f_a(x,y) = [\chi_{1,b}(a^{-1}x), y^c]$. f_a has the following properties: $f_a(x,1) = 1$, $f_a(a,b) = [b,b^c] \neq 1$, and for every $x \neq a$ we have $f_a(x,y) = 1$. Now the following $\cdot$ function will have the required properties:

$$\cdot(x,y) = \prod_{1\neq a\in G} p_{[b,b^c],a}(f_a(x,y)).$$

□

2.4 Semigroups

In this Section we determine the functionally complete semigroups and prove that every functionally complete semigroup is a group. Our leading reference on semigroups is [5]. We note that a semigroup does not necessarily have an identity element. For a semigroup $\mathbf{S}$ let $\mathbf{S}^1$ be the smallest semigroup which contains both an identity and $\mathbf{S}$. It is easy to see that $\mathbf{S}^1 = \mathbf{S}$ if $\mathbf{S}$ already contained and identity, otherwise $\mathbf{S}^1 = \mathbf{S} \cup \{1\}$. We remind the reader for the notion of $\mathcal{J}$-class:

Definition 27. Let $\mathbf{S}$ be a semigroup. Let us define the following relation: for every $a, b \in S$ we have $a \mathcal{J} b$ if and only if there exists $s_1, s_2, s_3, s_4 \in \mathbf{S}^1$, such that $b = s_1 a s_2$ and $a = s_3 b s_4$, i.e. a and b generate the same two-sided ideal. It is easy to verify that $\mathcal{J}$ is an equivalence relation. We call the classes of this equivalence relation $\mathcal{J}$-classes. Let J_a be the $\mathcal{J}$-class containing a.

There is a natural partial ordering on the $\mathcal{J}$-classes of a semigroup: let $J_a \leq J_b$ if and only if $\mathbf{S}^1 a \mathbf{S}^1 \subseteq \mathbf{S}^1 b \mathbf{S}^1$. We remind the reader that for any two elements a, b from the semigroup $\mathbf{S}$ we have that $J_{ab} \leq J_a$ and $J_{ab} \leq J_b$, moreover if $\mathbf{S}$ is finite then there is a unique minimal $\mathcal{J}$-class with respect to this ordering.

Theorem 28. *Every finite functionally complete semigroup is a group.*

Proof. Let $\mathbf{S}$ be a functionally complete semigroup, $|\mathbf{S}| \geq 2$. We first prove that $\mathbf{S}$ has only one $\mathcal{J}$-class. Indeed, let us suppose that $\mathbf{S}$ has at least two $\mathcal{J}$-classes. Let J_0 be the minimal $\mathcal{J}$-class by the usual ordering and let $s_1, s_2 \in S$ two elements of the semigroup such that $s_1 \in J_0$, $s_2 \notin J_0$. Let the function $f \colon S \to S$ be such that $f(s_1) = s_2$ and $f(s_2) = s_1$. This function cannot be realized by a polynomial over $\mathbf{S}$, as the semigroup multiplication is order-preserving, so is every polynomial but not the function f. Hence $\mathbf{S}$ has only one $\mathcal{J}$-class, which implies that $\mathbf{S}$ does not contain a 0 element with the property $0 \cdot s = s \cdot 0 = 0$ for every $s \in S$.

We conclude that $\mathbf{S}$ has exactly one $\mathcal{J}$-class, hence it is a Rees matrix semigroup without a 0, by the Rees – Suschkewitsch Theorem. Let I and J be the two index-sets, let $\mathbf{G}$ be the Schützenberger group and let $C \colon I \times J \to \mathbf{G}$ be the corresponding structure matrix. Now C contains elements only from $\mathbf{G}$ and does not contain 0. Now $\mathbf{S} = \mathcal{M}(\mathbf{G}; I, J; C)$ and $|\mathbf{S}| = |I| \cdot |J| \cdot |\mathbf{G}|$.

Let us suppose that $\mathbf{G} = \mathbf{1}$, then every entry of C is 1. If $|I| \cdot |J| \geq 3$, then we prove that $\mathbf{S}$ is not functionally complete. Without loss of generality we can assume that $|I| \geq |J|$. Let $i_0, i_1 \in I$ are two distinct elements. Let $\mathbf{S}'$ be the Rees matrix semigroup $\mathcal{M}(\mathbf{1}; I, J; C')$, where the Schützenberger group

is $\mathbf{1}$, the two index sets are $I' = I \setminus \{ i_0 \}$ (for an element $i_0 \in I$) and $J' = J$, and the structure matrix is $C' \colon I' \times J' \to \{ 1 \}$. Now clearly there exists a nontrivial homomorphism $h \colon \mathbf{S} \to \mathbf{S}'$, where $h([1; i_0, j]) = [1; i_1, j]$ and $h(s) = s$ otherwise. Hence $\mathbf{S}$ is not functionally complete by Proposition 4.

If $\mathbf{G} = \mathbf{1}$ and $|I| \cdot |J| = 2$, then $\mathbf{S}$ has the identity either $xy = y$ or $xy = x$. Let f be a unary function which interchanges the two elements of $\mathbf{S}$. Clearly, f can not be realized by a polynomial of $\mathbf{S}$, hence $\mathbf{S}$ is not functionally complete.

If $\mathbf{G} \neq \mathbf{1}$ and either $|I| \geq 2$ or $|J| \geq 2$, then with structure matrix $C' \colon I \times J \to \{ 1 \}$ we have that $h \colon \mathbf{S} \to \mathcal{M}(\mathbf{1}; I, J; C')$, $h([g; i, j]) = [1; i, j]$ is a nontrivial homomorphism and $\mathbf{S}$ is not functionally complete.

We can conclude that if $\mathbf{S}$ is functionally complete, then $\mathbf{S} \simeq \mathbf{G}$ and is a group. □

Remark 29. We note here that the proof actually shows that for finite semigroups the congruence-simple property is not equivalent with functional completeness as e.g. the two-element left-zero or two-element right-zero semigroups are congruence-simple but not functionally complete.

Corollary 30. *A finite semigroup* $\mathbf{S}$ *is functionally complete if and only if* $\mathbf{S}$ *is a finite simple non-Abelian group.*

2.5 Semirings

In the final Section of the Chapter we determine all functionally complete finite semirings and prove that all functionally complete semirings are rings. As in Section 2.4 we proved that every functionally complete semigroup is a group, we only consider functionally complete rings and groups later on in the thesis.

Some basic references on semirings are [9, 11, 18]. Semirings differ from rings only in that the addition is a commutative semigroup, not necessarily an Abelian group. These structures arise quite naturally as the endomorphisms of commutative semigroups.

Definition 31. A *semiring* $\mathbf{S} = (S, +, \cdot)$ is a nonempty set S with two associative operations $+$ and $\cdot$, the operation $+$ is commutative and both distributive laws holds, i.e. for every $a, b, c \in S$, $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ and $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$.

We later on omit the parentheses from $(a \cdot c) + (b \cdot c)$ and simply write $a \cdot c + b \cdot c$ as with rings. We note that sometimes the definition of a semiring includes a 0-element (identity for the addition). In that case, it is a requirement that

for every element s from the semiring $0 \cdot s = s \cdot 0 = 0$ applies. This always holds for rings, as the addition is an Abelian group. We, however, do not require that a semiring has a 0-element.

Theorem 32. *Every finite functionally complete semiring is a ring.*

Proof. Let $\mathbf{S} = (S, +, \cdot)$ be a functionally complete semiring. From Proposition 4 we know that it is congruence-simple. We claim first that the $\mathcal{J}$-class decomposition of the addition is a congruence. Indeed, let $a \sim b$ if and only if $a = b$ or if there exist $c, d \in S$ such that $b = a + c$ and $a = b + d$. It can be easily verified that if $a \sim b$, then for every $s \in S$ we have $a + s \sim b + s$, $s + a \sim s + b$, $a \cdot s \sim b \cdot s$ and $s \cdot a \sim s \cdot b$. As $\mathbf{S}$ is congruence-simple, $(S, +)$ has either only one $\mathcal{J}$-class, or every element of S is in a separate $\mathcal{J}$-class. We distinguish these two cases.

If every element is in a separate $\mathcal{J}$-class of $(S, +)$, then we define a partial ordering on the set S: let $a \leq b$ if and only if either $a = b$ or there exist $c \in S$ such that $a + c = b$. This partial ordering is the exact reverse of the usual $\mathcal{J}$-class ordering and if $a \leq b$ and $b \leq a$, then $a = b$ (since every element is in a separate $\mathcal{J}$-class). We claim that any polynomial p over $\mathbf{S}$ is order-preserving. For that we only have to prove that the basic operations $+$ and $\cdot$ are order-preserving.

The addition is order-preserving: if $a + c = b$, then for every x we have $a + x + c = a + c + x = b + x$, hence $a + x \leq b + x$. Similarly $x + a \leq x + b$ for every x if $a \leq b$. Finally if $a \leq b$ and $c \leq d$, then $a + c \leq a + d \leq b + d$.

Using the distributive law we can prove that the multiplication is order-preserving: if $a + c = b$, then $a \cdot x + c \cdot x = (a + c) \cdot x = b \cdot x$, hence $a \cdot x \leq b \cdot x$. Similarly if $a \leq b$, then $x \cdot a \leq x \cdot b$. Finally if $a \leq b$ and $c \leq d$, then $a \cdot c \leq a \cdot d \leq b \cdot d$.

Let $\alpha = \sum_{s \in S} s$ and let a be an arbitrary element different from α: $a \neq \alpha$. Now $a \leq \alpha$, for every polynomial p we have $p(a) \leq p(\alpha)$, therefore if a unary function $f \colon S \to S$ has the property $f(a) = \alpha$ and $f(\alpha) = a$, then f can not be realized by a polynomial over $\mathbf{S}$, since f is not order-preserving. This contradicts with our original assumption that $\mathbf{S}$ is functionally complete.

If $(S, +)$ has only one $\mathcal{J}$-class, then it is a (commutative) Rees matrix semigroup with no absorbing elements (an absorbing element forms a $\mathcal{J}$-class by itself), by the Rees–Susckewitsch Theorem [5]. Let I and J be the two index-sets, let $\mathbf{G}$ be the Schützenberger group and let $C \colon I \times J \to \mathbf{G}$ be the corresponding structure matrix. Now C contains elements only from $\mathbf{G}$ and $(S, +) = \mathcal{M}(\mathbf{G}; I, J; C)$. $(S, +)$ is commutative, hence for every $i_1, i_2 \in I$,

$j_1, j_2 \in J$, every $g_1, g_2 \in \mathbf{G}$ we have

$$\begin{aligned} [i_1, g_1, j_1]\,[i_2, g_2, j_2] &= [i_2, g_2, j_2]\,[i_1, g_1, j_1] \\ [i_1, g_1 C(j_1, i_2)\, g_2, j_2] &= [i_2, g_2 C(j_2, i_1)\, g_1, j_1] \end{aligned}$$

This means that $i_1 = i_2$ and $j_1 = j_2$, hence $|I| = |J| = 1$. Let $C(1,1) = h \in \mathbf{G}$. Now for every $g_1, g_2 \in \mathbf{G}$ we have $g_1 h g_2 = g_2 h g_1$. For $g_2 = 1$ we have $g_1 h = h g_1$ for every $g_1 \in \mathbf{G}$, hence $h \in Z(\mathbf{G})$. From that we conclude to $g_1 g_2 = g_2 g_1$ for every $g_1, g_2 \in \mathbf{G}$, hence $\mathbf{G}$ is Abelian. Now $\varphi\colon (S, +) \to \mathbf{G}$, $\varphi([1, g, 1]) = gh$ is an isomorphism between $\mathbf{G}$ and $(S, +)$: $\varphi([1, g_1, 1]\,[1, g_2, 1]) = \varphi([1, g_1 h g_2, 1]) = g_1 h g_2 h = \varphi([1, g_1, 1])\,\varphi([1, g_2, 1])$. Therefore $(S, +)$ is an Abelian group and $\mathbf{S}$ is a ring. □

Remark 33. We note here that for finite semirings the congruence-simple property is not equivalent with the functional completeness. Let $V(\mathbf{G})$ be the following semiring for any group $\mathbf{G}$: the underlying set is $G \cup \{\infty\}$; the multiplication is the group multiplication, $x \cdot \infty = \infty \cdot x = \infty$; and for the addition $x + x = x$ and $x + y = \infty$ for $x \neq y$. It is easy to see that $V(\mathbf{G})$ is congruence simple for any finite group, but not functionally complete.

Corollary 34. *A finite semiring is functionally complete if and only if it is a matrix ring over a finite field.*

Chapter 3

Length of polynomial expressions

In Chapter 2 we determined functionally complete algebras for several classes. Moreover, our proofs are all algorithmic, so they all give us some method to realize arbitrary functions over these functionally complete algebras. There are of course many different realizations of a function. One usually wants to find an optimal realization (or one close to the optimal) in some sense, e.g. a polynomial which can be calculated efficiently. Efficiency can, however, be measured in may ways. In this Chapter we consider one of the most basic ones: the length of the realizing polynomials. We give upper and lower bounds on the length of arbitrary functions over arbitrary and specific functionally complete algebras. We consider computational models in Chapter 4.

We define the length of a *polynomial expression* over an algebra $\mathbf{A} = (A, f_1, \ldots, f_k)$ (i.e. an expression which can be composed from variables, the basic operations and some constants from A) in a natural way. We give a definition which represents the idea that the length of a polynomial p is the number of occurrences of the constants and the variables p has. This definition coincides with the usual length definition for group polynomials. Denote the length of a polynomial expression $p(x_1, \ldots, x_n)$ with $\|p(x_1, \ldots, x_n)\|$.

Definition 35. The *length* of a polynomial expression over $\mathbf{A}$ is defined recursively:

1. The length of a variable x or a constant c is 1: $\|x\|_{\mathbf{A}} = \|c\|_{\mathbf{A}} = 1$.

2. For an m-variable basic function f of $\mathbf{A}$ and for polynomial expressions $p_1, \ldots, p_m$, the length of $f(p_1, \ldots, p_m)$ is the sum of the lengths of the p_i's: $\|f(p_1, \ldots, p_m)\|_{\mathbf{A}} = \sum_{i=1}^{m} \|p_i\|_{\mathbf{A}}$. Then the length of $f(x_1, \ldots, x_m)$ is $\|f\|_{\mathbf{A}} = m$.

We usually omit the subscript and just write $\|p\|$ for the length of a polynomial.

We have to mention here that every polynomial has a corresponding rooted tree with ordered edges and labelled nodes. Every inner node represents a basic function in the polynomial, the children of a node represent the inputs of the corresponding basic function in the polynomial. The ordering of the edges determines the ordering of the inputs. Finally the leaves represent constants and variables. From now on by an edge uv we mean an edge, where v is a child of u.

Now length can be defined by using this rooted tree. Let the length of the polynomial be the number of leaves in the corresponding rooted tree. It is easy to check that the length by this definition is exactly the same as by Definition 35. Technically the length of a polynomial is the number of occurrences of constants and variables in p (counting multiplicities).

Another definition could be to define the length of a polynomial as the number of inner nodes in the corresponding tree (as in e.g. [40]). Technically this definition counts the number of the basic functions used. This definition of length is almost the same as ours, apart from the use of the basic unary operations. Generally the 'unary part' of the algebra is not really interesting, as by the composition of unary functions we only obtain unary functions. This idea suggests the notion of the *branching tree*: we take the usual rooted tree corresponding to a polynomial and collapse every chain of unary operations into a single edge. Then we label the edges with a unary polynomial which we obtain by composing the unary basic operations of the corresponding chain. The precise definition is the following:

Definition 36. For every polynomial p we define the corresponding *branching tree*. The branching tree has one root with degree exactly one. For a branching tree T we denote this root by r_T and we denote the label of the edge of r_T by e_T. The tree is defined recursively:

1. The branching tree T for a variable x_i has two nodes u and $r = r_T$, and an edge ru. The node u is labelled by x_i and the edge ru is labelled by $e_T = id$ (the identity unary operation).

2. The branching tree T for a constant c has two nodes u and r, and an edge ru. The node u is labelled by c and the edge ru is labelled by $e_T = id$ (the identity unary operation).

3. Let f be a unary basic operation and let p be a polynomial with branching tree T. Now the branching tree T' of the polynomial $f(p)$ is the same as T, except that $e_{T'}$ is the polynomial $g \circ e_T$.

4. Let f be a k-ary basic operation ($k \geq 2$) and let $p_1, \ldots, p_k$ be polynomials with branching trees $T_1, \ldots, T_k$. Now the branching tree T' of the polynomial $f(p_1, \ldots, p_k)$ is constructed by identifying $r_{T_1}, \ldots, r_{T_k}$ into a single node u, labelling it by f, adding a root r_T with an edge $r_T u$ and labelling this by id (the identity unary operation). When we identify the nodes r_{T_i} we linearly order the edges of u. The ordering will be that the edge uv is the ith if v was originally in the tree T_i.

Therefore in the corresponding branching tree the edges and nodes are labelled, moreover the edges have a numbering. Every inner node (non-leaf and non-root node) represents a basic non-unary function in the polynomial, and the children of a node represent the inputs of the corresponding basic function in the polynomial. The ordering of the edges at a node determines the ordering of the inputs of the corresponding basic function. The leaves represent constants and variables. Finally if v is a child of u, then the labelling of the edge uv represents the composition of basic unary functions (or the identity), which is applied on the result of v in the polynomial p. If $\mathbf{A}$ is a group, then every edge uv (v is a child of u) is labelled either with the $^{-1}$ or with the identity, depending on whether we invert the result of v before applying u in the polynomial p. Similarly if $\mathbf{A}$ is the two-element Boolean algebra $\mathbf{B}$, then every edge is labelled either with $\neg$ or with the identity, depending on whether there is a negation at that particular place in the polynomial p.

It is easy to see that the number of inner nodes in the corresponding branching tree is essentially the same as the length of the polynomial (the difference is 1). Moreover, if the algebra has no unary basic operations then the corresponding branching tree is essentially the same as the usual corresponding rooted tree. This is the case in [40], where the two-element algebra with 16 binary basic operations was considered.

In this Chapter we search for short polynomials realizing particular functions. In many cases we denote a function and its realizing polynomial with the same symbol. In most of the cases this polynomial is a shortest one. In order not to create confusion we introduce the following definition:

Definition 37. The length of a function f over an algebra $\mathbf{A}$ is the length of a shortest polynomial p over $\mathbf{A}$ realizing the function f. We denote the length of f with $\|f\|_{\mathbf{A}}$ or shortly $\|f\|$:

$$\|f\|_{\mathbf{A}} = \min \{ \|p\|_{\mathbf{A}} : p \text{ realizes } f \text{ over } \mathbf{A} \}.$$

Throughout the Chapter we plan to give upper and lower bounds for the length of polynomials realizing arbitrary functions. We calculate these

bounds for arbitrary and for specific functionally complete algebras, then we compare the results.

For several algebras the length of a polynomial is closely related to the number of variables in the polynomial expression (including multiplicities). Therefore we introduce the following notion: Let $v_{\mathbf{A}}(p)$ (or shortly $v(p)$) be the number occurrences of the variables (counting multiplicities) in the polynomial expression p containing n variables $x_1, \ldots, x_n$. Later we might use the term '*number of variable occurrences*' as well. Let $v_i(p)$ be the number occurrences of the ith variable x_i (counting multiplicities) in the polynomial expression p. If p is an n-ary polynomial expression then $v(p) = \sum_{i=1}^{n} v_i(p)$. Similarly to $\|f\|$ we define $v_{\mathbf{A}}(f)$ (shortly $v(f)$) for a function f:

$$v_{\mathbf{A}}(f) = \min\{\, v_{\mathbf{A}}(p) : p \text{ realizes } f \text{ over } \mathbf{A} \,\}.$$

Remark 38. We do not know whether for every functionally complete algebra and for any arbitrary function $f\colon A^n \to A$ there exists a polynomial p over $\mathbf{A}$ such that $\|f\| = \|p\|$ and $v(f) = v(p)$. This is not true for *partial functions* (see Remark 81). We do know that $\|p\| = \|f\|$ does not imply $v(p) = v(f)$: the polynomials $x + x$ and $2 \cdot x$ realize the same function over the finite ring $\mathbf{Z}_5$, they both have length two (which is the length of the function), but only one of them has one variable occurrences.

Now we mention some properties of the length and the number of variable occurrences. An immediate consequence of the definition are the following lemmas:

Lemma 39. *For polynomial expressions $p, q_1, \ldots, q_n$ we have that*

$$\|p(q_1, \ldots, q_n)\| \leq \|p\| \cdot \max\{\, \|q_i\| : i = 1, \ldots, n \,\}.$$

Proof. Let q be a polynomial from $\{\, q_1, \ldots, q_n \,\}$, for which the length is maximal: $\|q\| = \max\{\, \|q_i\| : 1 \leq i \leq n \,\}$. Then

$$\|p(q_1, \ldots, q_n)\| = \sum_{i=1}^{n} \|q_i\| \leq n \cdot \|q\| = \|p\| \cdot \|q\|.$$

□

Lemma 39 holds for the number of variable occurrences, too:

Lemma 40. *For polynomial expressions $p, q_1, \ldots, q_n$ we have that*

$$v(p(q_1, \ldots, q_n)) \leq v(p) \cdot \max\{\, v(q_i) : i = 1, \ldots, n \,\}.$$

Proof. Let q be a polynomial from $\{ q_1, \ldots, q_n \}$, for which the number of variable occurrences is maximal: $v(q) = \max \{ v(q_i) : 1 \leq i \leq n \}$. When substituting the variable x_i with q_i in the expression p, then every variable is substituted by a polynomial expression with at most $v(q)$ variable occurrences. Hence there are at most $v(p) \cdot v(q)$ variable occurrences in the expression $p(q_1, \ldots, q_n)$. □

With a slight modification of the proof we have the following:

Lemma 41. *For polynomial expressions $p, q_1, \ldots, q_n$ we have that*

$$\|p(q_1, \ldots, q_n)\| \leq \|p\| + \sum_{i=1}^{n} v_i(p) \cdot (\|q_i\| - 1).$$

Proof. When substituting the ith variable by q_i in the expression p, then length increases at most by $v_i(p) \cdot (\|q_i\| - 1)$. Hence comparing with $\|p\|$ the length increases by at most $\sum_{i=1}^{n} v_i(p) \cdot (\|q_i\| - 1)$ which finishes the proof. □

Corollary 42. *For polynomial expressions $p, q_1, \ldots, q_n$ we have that*

$$\|p(q_1, \ldots, q_n)\| \leq \|p\| + v(p) \cdot \max \{ \|q_i\| - 1 : i = 1, \ldots, n \}.$$

Remark 43. It is easy to see that Lemma 39 and Lemma 40 hold not only for polynomial expressions but for functions, too. Lemma 41 and Corollary 42, however, may not necessarily hold for functions: it might happen that for some function f the length $\|f\|$ and $v(f)$ cannot be realized by the same polynomial expression.

Let us recall the proof of Theorem 6, where we mentioned that both $\prod$ and $\sum$ can be defined in an arbitrarily iterated way (even if $\cdot$ and $+$ are not associative in general). The following lemma is one of the most important lemmas in this Chapter. Here we give a fast and short method for some iterations, which we use later on as well. From now on by log we mean the base 2 logarithm function.

Lemma 44. *Let p be a binary polynomial over an algebra $\mathbf{A}$. Define the following polynomial expressions: $p^{(1)}(x_1) = x_1$, $p^{(2)}(x_1, x_2) = p(x_1, x_2)$ and for every integer $n \geq 2$:*

$$\begin{aligned} p^{(2n-1)}(x_1, \ldots, x_{2n-1}) &= p\left(p^{(n)}(x_1, \ldots, x_n), p^{(n-1)}(x_{n+1}, \ldots, x_{2n-1})\right) \\ p^{(2n)}(x_1, \ldots, x_{2n}) &= p\left(p^{(n)}(x_1, \ldots, x_n), p^{(n)}(x_{n+1}, \ldots, x_{2n})\right). \end{aligned}$$

Let $v\left(p^{(n)}\right)$ be the number of variable occurrences in $p^{(n)}$, let $\left\|p^{(n)}\right\|$ be the length of $p^{(n)}$ and let $V = v\left(p^{(2)}\right)$, $L = \left\|p^{(2)}\right\|$. Suppose that $V \geq 2$. Then

$$\begin{aligned} v\left(p^{(n)}\right) &< V \cdot n^{\log V}, && (3.1)\\ \left\|p^{(n)}\right\| &= \frac{L-1}{V-1} \cdot \left(v\left(p^{(n)}\right) - 1\right) + 1, && (3.2)\\ \left\|p^{(n)}\right\| &< \frac{L-1}{V-1} \cdot \left(V \cdot n^{\log V} - 1\right) + 1, && (3.3)\\ \left\|p^{(n)}\right\| &< L \cdot n^{\log L}, && (3.4)\\ \left\|p^{(n)}\right\| &< 2 \cdot L \cdot n^{\log V}. && (3.5) \end{aligned}$$

Proof. Let V_i be the number of variable x_i occurrences in p, $V = V_1 + V_2$. By the definition of $p^{(n)}$ and using the ideas of proof of Lemma 40 and Lemma 41 we can give an easy recursion for $v\left(p^{(n)}\right)$ and $\left\|p^{(n)}\right\|$: $v\left(p^{(1)}\right) = \left\|p^{(1)}\right\| = 1$, $v\left(p^{(2)}\right) = V$, $\left\|p^{(2)}\right\| = L$ and for every $n \geq 2$:

$$\begin{aligned} v\left(p^{(2n-1)}\right) &= V_1 \cdot v\left(p^{(n)}\right) + V_2 \cdot v\left(p^{(n-1)}\right)\\ v\left(p^{(2n)}\right) &= V \cdot v\left(p^{(n)}\right),\\ \left\|p^{(2n-1)}\right\| &= \left\|p^{(2)}\right\| + V_1 \cdot \left(\left\|p^{(n)}\right\| - 1\right) + V_2 \cdot \left(\left\|p^{(n-1)}\right\| - 1\right)\\ \left\|p^{(2n)}\right\| &= \left\|p^{(2)}\right\| + V \cdot \left(\left\|p^{(n)}\right\| - 1\right). \end{aligned}$$

Solving the recursion is usually hard, but we can estimate using some properties of $v\left(p^{(n)}\right)$ and $\left\|p^{(n)}\right\|$.

First we prove (3.2) by induction on n. The equation (3.2) holds for $n = 1$ and for $n = 2$, since $\left\|p^{(1)}\right\| - 1 = 0 = \frac{L-1}{V-1} \cdot \left(v\left(p^{(1)}\right) - 1\right)$ and $\left\|p^{(2)}\right\| - 1 = L - 1 = \frac{L-1}{V-1} \cdot (V-1) = \frac{L-1}{V-1} \cdot \left(v\left(p^{(2)}\right) - 1\right)$. Let us suppose that (3.2) holds for $k < 2n-1$ for some $n \geq 2$ and check $\left\|p^{(2n-1)}\right\| - 1$:

$$\begin{aligned} \left\|p^{(2n-1)}\right\| - 1 = V_1 \cdot \left(\left\|p^{(n)}\right\| - 1\right) + V_2 \cdot \left(\left\|p^{(n-1)}\right\| - 1\right) + L - 1 =\\ V_1 \cdot \left(\frac{L-1}{V-1} \cdot \left(v\left(p^{(n)}\right) - 1\right)\right) + V_2 \cdot \left(\frac{L-1}{V-1} \cdot \left(v\left(p^{(n-1)}\right) - 1\right)\right) + L - 1 =\\ \frac{L-1}{V-1}\left(V_1 v\left(p^{(n)}\right) + V_2 v\left(p^{(n-1)}\right)\right) + (L-1) \cdot \left(1 - \frac{V_1+V_2}{V-1}\right) =\\ \frac{L-1}{V-1} \cdot \left(v\left(p^{(2n-1)}\right) - 1\right). \end{aligned}$$

Now let us suppose that (3.2) holds for $k < 2n$ for some $n \geq 2$ and check

$\left\|p^{(2n)}\right\| - 1$:

$$\begin{aligned}\left\|p^{(2n)}\right\| - 1 &= V \cdot \left(\left\|p^{(n)}\right\| - 1\right) + L - 1 = \\ &= V \cdot \frac{L-1}{V-1} \cdot \left(v\left(p^{(n)}\right) - 1\right) + L - 1 = \\ &= \frac{L-1}{V-1} \cdot Vv\left(p^{(n)}\right) + (L-1) \cdot \left(1 - \frac{V}{V-1}\right) = \\ &= \frac{L-1}{V-1} \cdot \left(v\left(p^{(2n)}\right) - 1\right).\end{aligned}$$

This finishes the proof of (3.2).

We continue by proving (3.1). We claim first that $v\left(p^{(n)}\right)$ is strictly monotone in n, i.e. $v\left(p^{(n)}\right) < v\left(p^{(n+1)}\right)$ for every positive integer n. We prove this statement by induction on n. The statement is clearly true for $n = 1$, since $v\left(p^{(2)}\right) = V \geq 2 > 1 = v\left(p^{(1)}\right)$. Suppose that $v\left(p^{(k-1)}\right) < v\left(p^{(k)}\right)$ for every $k < 2n-1$ (for some $n \geq 2$) and let us check $v\left(p^{(2n-1)}\right)$:

$$v\left(p^{(2n-1)}\right) = V_1 \cdot v\left(p^{(n)}\right) + V_2 \cdot v\left(p^{(n-1)}\right) > V \cdot v\left(p^{(n-1)}\right) = v\left(p^{(2n-2)}\right).$$

Now suppose that $v\left(p^{(k-1)}\right) < v\left(p^{(k)}\right)$ and for every $k < 2n$ (for some $n \geq 2$) and let us check $v\left(p^{(2n)}\right)$:

$$v\left(p^{(2n)}\right) = V \cdot v\left(p^{(n)}\right) > V_1 \cdot v\left(p^{(n)}\right) + V_2 \cdot v\left(p^{(n-1)}\right) = v\left(p^{(2n-1)}\right).$$

Thus $v\left(p^{(n)}\right)$ is strictly monotone in n.

For $n = 2^k$ we can calculate $v\left(p^{(n)}\right)$:

$$v\left(p^{(n)}\right) = v\left(p^{(2^k)}\right) = V \cdot v\left(p^{(2^{k-1})}\right) = \cdots = V^k = n^{\log V}.$$

Now let n be arbitrary and suppose that $2^{k-1} \leq n < 2^k$. Then

$$v\left(p^{(n)}\right) < v\left(p^{(2^k)}\right) = \left(2^k\right)^{\log V} \leq (2n)^{\log V} = V \cdot n^{\log V}.$$

The inequality (3.3) immediately follows from (3.2) and (3.1). For proving the inequality (3.4) we mention that $\frac{V^{1+\log n}-1}{V-1}$ is strictly monotone in V for $n \geq 2$. If we we change V to L in (3.3) then we can only increase the value of the righthand side and we have

$$\left\|p^{(n)}\right\| - 1 < \frac{L-1}{L-1} \cdot \left(L \cdot n^{\log L} - 1\right) = L \cdot n^{\log L} - 1.$$

Finally (3.5) follows from (3.3) and the fact that $\frac{V}{V-1} \leq 2$. □

Now, closely examining the proofs of Theorem 6 and Theorem 9, recalling Remark 10, then applying Lemma 39 and Lemma 44 we obtain the following for a functionally complete algebra $\mathbf{A}$ with two arbitrary distinct elements 0 and 1:

Theorem 45. *Let $\mathbf{A}$ be a functionally complete algebra. Let $0, 1 \in A$ be two distinct elements and let $+$, $\cdot$ and χ_a be shortest polynomials with properties such as in Theorem 6. Let T be any positive real number for which $T \geq \max\{\, \|\chi_a\| : a \in A \,\}$. Let d be a shortest discriminator polynomial. Let $\chi_{a_1,\ldots,a_n}$ be a shortest characteristic polynomial for the n-tuple $(a_1, \ldots, a_n)$. Let $-$ be a shortest polynomial such that $x - y = 0$ if and only if $x = y$ and $1 - 0 = 1$. Let p be a shortest polynomial realizing an arbitrary n-ary function f over $\mathbf{A}$ with e-many non-zero values, where $1 \leq e \leq |A|^n$. Then the following inequalities hold:*

$$\|\chi_a\| \leq \|-\| \cdot \|\chi_0\| \tag{3.6}$$

$$\|\chi_{a_1,\ldots,a_n}\| \leq 2 \cdot \|\cdot\| \cdot n^{\log v(\cdot)} \cdot \max\{\, \|\chi_{a_i}\| : 1 \leq i \leq n \,\}, \tag{3.7}$$

$$\|p\| \leq 2 \cdot \|\cdot\| \cdot \|+\| \cdot e^{\log v(+)} \cdot \max\{\, \|\chi_{a_1,\ldots,a_n}\| : a_i \in A \,\}, \tag{3.8}$$

$$\|p\| \leq 4 \cdot \|\cdot\|^2 \cdot \|+\| \cdot e^{\log v(+)} \cdot n^{\log v(\cdot)} \cdot T, \tag{3.9}$$

$$\|p\| \leq 4 \cdot \|-\| \cdot \|\cdot\|^2 \cdot \|+\| \cdot e^{\log v(+)} \cdot n^{\log v(\cdot)} \cdot \|\chi_0\|, \tag{3.10}$$

$$\|d\| \geq \max\left\{ \|+\|, \|\chi_0\|, \sqrt{\|\cdot\|}, \sqrt{\|\chi_a\|} \right\}, \tag{3.11}$$

$$\|d\| \leq \|-\|^2 \cdot \|+\| \cdot \|\cdot\| \cdot \|\chi_0\|. \tag{3.12}$$

If $|A| = 2$, then

$$\|p\| \leq 2 \cdot \|+\| \cdot e^{\log v(+)} \cdot \max\{\, \|\chi_{a_1,\ldots,a_n}\| : a_i \in A \,\}, \tag{3.13}$$

$$\|p\| \leq 4 \cdot \|\cdot\| \cdot \|+\| \cdot e^{\log v(+)} \cdot n^{\log v(\cdot)} \cdot T, \tag{3.14}$$

$$\|p\| \leq 4 \cdot \|-\| \cdot \|\cdot\| \cdot \|+\| \cdot e^{\log v(+)} \cdot n^{\log v(\cdot)} \cdot \|\chi_0\|. \tag{3.15}$$

Proof. The proof is simply applying Lemma 39 and Lemma 44 on the following representations:

$$\begin{aligned}
\chi_a(x) &= \chi_0(x - a), \\
\chi_{a_1,\ldots,a_n}(x_1, \ldots, x_n) &= \prod_{i=1}^{n} \chi_{a_i}(x_i), \\
p(x_1, \ldots, x_n) &= \sum_{(a_1,\ldots,a_n) \in A^n} \left(p(a_1, \ldots, a_n) \cdot \chi_{a_1,\ldots,a_n}(x_1, \ldots, x_n) \right), \\
d(x, y, z) &= z \cdot \chi_0(x - y) + x \cdot (1 - \chi_0(x - y)).
\end{aligned}$$

Here we consider $\prod$ and $\sum$ as the iterated versions of $\cdot$ and $+$ in the way described in Lemma 44. If $|A| = 2$, then p can be represented as

$$p(x_1, \ldots, x_n) = \sum_{(a_1,\ldots,a_n)\in A^n} \chi_{a_1,\ldots,a_n}(x_1, \ldots, x_n).$$

□

Checking Theorem 45, especially (3.9) we understand the importance of the number of occurrences of $+$ and $\cdot$. The number e usually has the order of $|A|^n$, hence e is very large compared to the other elements occurring in (3.9). In our bound e is taken to the power $\log v(+)$, so we can obtain the best bound when $+$ is a binary polynomial with only two variable occurrences, i.e. $+$ is basically a basic operation of the algebra $\mathbf{A}$. This is the case for Boolean algebras, rings or for groups. The following theorem shows that it is essential that a bound has a such large factor:

Theorem 46. *Let $\mathbf{A}$ be a functionally complete algebra. For every $\varepsilon > 0$ and for sufficiently large n (depending on ε) there exists an n-ary function f over $\mathbf{A}$, such that*

$$\|f\| \geq \frac{\log|A|}{1+\varepsilon} \cdot \frac{|A|^n}{\log n}.$$

Proof. We use the same counting idea as e.g. [40], which shows a similar bound for the two-element algebra with 16 binary operations as basic operations. Let us consider the number of functions *f which can be realized by a polynomial with length at most l*. This way we count the number of functions for which $\|f\| \leq l$. Let this number be $N(l)$. If L is the least length such that all n-ary functions have length at most L, then $N(L) \geq |A|^{|A|^n}$. This gives us a lower bound on the length.

Let $\mathbf{A}$ have m-many basic operation symbols, which are neither unary operation symbols nor constant symbols. Let us suppose that every basic operation of $\mathbf{A}$ is at most k-ary. For an arbitrary polynomial p with $\|p\| \leq l$ we consider the corresponding branching tree as we defined it in Definition 35. In this tree every inner node (non-leaf and non-root node) represents a basic non-unary function in the polynomial and the children of a node represent the inputs of the corresponding basic function in the polynomial. The labelling of the edges corresponds to some composition of the basic unary operations. The branching tree has at most l leaves for every polynomial p with $\|p\| \leq l$.

There are at most $l-1$ inner nodes in the tree, all of them are chosen from m-many (at least binary) basic operations, so for every tree the labelling of the inner nodes can be done at most m^{l-1}-many ways. Each leaf is either a

variable or a constant, so all leaves can be labelled at most $(n+|A|)^l$-many ways. Moreover there are at most $\left(3^k\right)^{l-1}$-many labellings for nodes and leaves of the trees corresponding to polynomials $\|p\| \leq l$: every inner node has k possible children and for each possible child there are three options (the child is a leaf, the child is another inner node, or the child does not exist).

We have not estimated on the labelling of the edges, yet. There exist potentially infinitely many possible labellings of an edge, but every label realizes one of the $|A|^{|A|}$-many unary functions. If two branching trees differ only by a label of an edge, and both labels represent the same function, then the corresponding polynomials represent the same function, too. Therefore we count the possible realizations of every edge-label. Every edge is labelled by a unary function, thus the edges can be labelled in at most $\left(|A|^{|A|}\right)^{2l-1}$-many ways, as there are at most $2l-1$-many edges. Hence

$$N(l) \leq (n+|A|)^l \cdot m^{l-1} \cdot 3^{k\cdot(l-1)} \cdot |A|^{|A|\cdot 2l}.$$

Let f be a longest n-ary function. Let $L = \|f\|$. Now applying $N(L) \geq |A|^{|A|^n}$ we have that

$$|A|^n \cdot \log|A| \leq (L-1)\cdot(\log m + k\cdot\log 3) + L\cdot(\log(n+|A|) + 2\,|A|\cdot\log|A|).$$

Let us fix an $\varepsilon > 0$. For sufficiently large n we have $|A| \leq n$, thus

$$|A|^n \cdot \log|A| \leq L\cdot(\log m + k\cdot\log 3 + 1 + \log n + 2\,|A|\cdot\log|A|).$$

For sufficiently large n we have $\log m + k\cdot\log 3 + 1 + 2\,|A|\cdot\log|A| \leq \varepsilon\cdot\log n$, therefore we obtain

$$\|f\| = L \geq \frac{\log|A|}{1+\varepsilon}\cdot\frac{|A|^n}{\log n}.$$

□

Remark 47. Though slightly sharper lower bounds can be derived for particular algebras (as e.g. in [40] for the two-element algebra with all 16 binary basic operations), we do not calculate those here explicitly.

The other important factor in the upper bound (3.9) is n taken to the power $\log v(\cdot)$. We can get rid of this factor for some special algebras:

Theorem 48. *Let* $\mathbf{A}$ *be a functionally complete algebra,* $N = |A|$. *Let* $0, 1 \in A$ *be two distinct elements and let* $+$, $\cdot$ *and* χ_a *be shortest polynomials with properties such as in Theorem 6 and let us suppose that* $\|+\| = v(+) =$

$v(\cdot) = \|\cdot\| = 2$. Let p be a shortest polynomial realizing an arbitrary n-ary function f over $\mathbf{A}$ with e-many non-zero values, where $1 \leq e \leq |A|^n$. Let T be any positive real number for which $T \geq \max\{\, \|\chi_a\| : a \in A \,\}$. Then the following inequalities hold if $N \geq 3$:

$$\|p\| \leq e \cdot (1 + T \cdot (3 + n - \log_N e)) - 2 \cdot T,$$
$$\|p\| \leq e \cdot \left(1 + T \cdot \left(3 + n - \frac{\log e}{\log N}\right)\right) - 2 \cdot T,$$

If $N = 2$, then

$$\|p\| \leq ((3 + n - \log e) \cdot e - 2) \cdot T.$$

Proof. Consider the case where $|A| \geq 3$. The second inequality is the same as the first one using $\log_N e = \frac{\log e}{\log N}$. We prove the first inequality by induction on n. If $n = 1$, then $f(x) = \sum_{a \in A} f(a) \cdot \chi_a(x)$, which has length at most $e \cdot (1 + T) \leq e \cdot (1 + T \cdot (3 + 1 - \log_N e)) - 2 \cdot T$ if we do not put any of those summands into the polynomial where $f(a) = 0$.

The idea of the proof is that we try to calculate f recursively. For every element $a \in A$ let f_a be an $n-1$-ary function, such that $f_a(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, a)$. Now $f(x_1, \ldots, x_n) = \sum_{a \in A} f_a(x_1, \ldots, x_{n-1}) \cdot \chi_a(x_n)$. Let f_a have e_a-many non-zero values. Now we apply the induction hypothesis for the $n-1$-ary functions. If there is only one $e_a > 0$, then $e_a = e$ and

$$\begin{aligned}\|p\| &\leq \|f_a\| + T \leq e \cdot (1 + T \cdot (3 + n - 1 - \log_N e)) - 2 \cdot T + T \\ &\leq e \cdot (1 + T \cdot (3 + n - \log_N e)) - 2 \cdot T.\end{aligned}$$

Otherwise

$$\begin{aligned}\|p\| &\leq \sum_{a \in A} (\|f_a\| + T) \\ &\leq \sum_{e_a > 0} (e_a \cdot (1 + T \cdot (3 + n - 1 - \log_N e_a)) - 2 \cdot T + T) \\ &\leq (e_0 + e_1) \cdot (3 + n) - T \cdot (e_0 + e_1 + e_0 \cdot \log e_0 + e_1 \cdot \log e_1) - 2T \\ &\leq \sum_{e_a > 0} e_a \cdot (1 + T \cdot (3 + n)) - T \cdot \left(\sum_{e_a > 0} e_a + \sum_{e_a > 0} e_a \log_N e_a\right) - \sum_{e_a > 0} T \\ &= e \cdot (1 + T \cdot (3 + n - \log_N e)) - 2 \cdot T.\end{aligned}$$

The last of these inequalities holds by the following lemma:

Lemma 49. *Let $N \geq 2$ be a positive natural number, let $k \leq N$ be a positive natural number, too. If $e_1, \ldots, e_k$ are positive real numbers, $e = \sum_{i=1}^{k} e_i$, then*

$$e \cdot \log_N e \leq e + \sum_{i=1}^{k} (e_i \cdot \log_N e_i).$$

Proof. The function $f\colon \mathbb{R}^+ \to \mathbb{R}$, $f(x) = x \cdot \log_N x$ is convex as the second derivative is positive. Therefore we have

$$\begin{aligned}
\frac{\sum_{i=1}^{k} e_i}{k} \cdot \log_N \frac{\sum_{i=1}^{k} e_i}{k} &\leq \frac{\sum_{i=1}^{k} e_i \cdot \log_N e_i}{k}, \\
e \cdot (\log_N e - \log_N k) &\leq \sum_{i=1}^{k} e_i \cdot \log_N e_i, \\
e \cdot (\log_N e - 1) &\leq \sum_{i=1}^{k} e_i \cdot \log_N e_i, \\
e \cdot \log_N e &\leq e + \sum_{i=1}^{k} (e_i \cdot \log_N e_i).
\end{aligned}$$

□

The $N = 2$ case differs only in that we do not need to multiply by a constant, since the only constant differing from 0 is 1. Hence if $n = 1$, then $f(x) = \sum_{a \in A, f(a) \neq 0} \chi_a(x)$, which has length at most

$$e \cdot T \leq ((3 + 1 - \log e) \cdot e - 2) \cdot T.$$

The induction goes the same way as with the case of $N \geq 3$, and we have

$$\begin{aligned}
\|p\| &\leq (\|f_0\| + T) + (\|f_1\| + T) \\
&\leq ((3 + n - 1 - \log e_0) \cdot e_0 - 2 + 1) \cdot T \\
&+ ((3 + n - 1 - \log e_1) \cdot e_1 - 2 + 1) \cdot T \\
&\leq e \cdot (1 + T \cdot (3 + n)) - T \cdot e \cdot \log_N e - 2 \cdot T \\
&\leq ((3 + n - \log e) \cdot e - 2) \cdot T,
\end{aligned}$$

if both e_0 and e_1 are positive. If one of them is 0, then

$$\begin{aligned}
\|p\| &\leq ((3 + n - 1 - \log e) \cdot e - 2 + 1) \cdot T \\
&\leq ((3 + n - \log e) \cdot e - 2) \cdot T.
\end{aligned}$$

□

Remark 50. The idea of building up a polynomial recursively only works when the operations $+$ and $\cdot$ have the shortest possible representations, which means that they are quite close to some basic binary functions of $\mathbf{A}$. If either $+$ or $\cdot$ has more than two variable occurrences then mixing them up will end up having another exponential factor in the bound, which we wanted to get rid of. Theorem 48 is useful for calculating bounds for the two-element Boolean algebra or for functionally complete rings as we see it in Section 3.2 and in Section 3.3.

Remark 51. Notice that comparing the result (3.9) in Theorem 45 with the result of Theorem 48 we almost completely got rid of the factor n. We have a factor $(3 + n - \log_N e)$ instead, but if e is large then this factor is just a constant, e.g. if $e = c_1 \cdot N^{n-c_2}$, then this factor is at most $(3 + c_2 - \log_N c_1)$. If, on the other hand, e is really small, e.g. $e = c_3 \cdot N^{c_4 \cdot n}$ where $c_4 < 1$, then this factor turns out to be linear in n: $(1 - c_4) \cdot n + (3 - \log_N c_3)$. In that case e being small compensates for the slightly larger second factor, so we do not lose anything (compared to Theorem 45) by having the factor $(3 + n - \log_N e)$.

Finally we summarize the upper and lower bounds:

Corollary 52. *Let* $\mathbf{A}$ *be a functionally complete algebra, let* $N = |A|$. *Let* $0, 1 \in A$ *be two distinct elements and let* $+$, $\cdot$ *and* χ_a *be shortest polynomials with properties such as in Theorem 6. Let* T *be any positive real number for which* $T \geq \max\{\, \|\chi_a\| : a \in A \,\}$. *Let* f *be an arbitrary* n*-ary function over* $\mathbf{A}$. *Then*

$$\|f\| \leq 4 \cdot \|\cdot\|^2 \cdot \|+\| \cdot \left(N^{\log v(+)}\right)^n \cdot n^{\log v(\cdot)} \cdot T,$$

If $N = 2$ *then we can replace the factor* $\|\cdot\|^2$ *by* $\|\cdot\|$.
If $\|+\| = v\,(+) = \|\cdot\| = v\,(\cdot) = 2$ *then*

$$\|f\| \leq (3T + 1) \cdot N^n - 2 \cdot T.$$

If $\|+\| = v\,(+) = \|\cdot\| = v\,(\cdot) = N = 2$ *then*

$$\|f\| \leq (3N^n - 2) \cdot T.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_0 *over* $\mathbf{A}$ *such that*

$$\|f_0\| \geq \frac{\log N}{1 + \varepsilon} \cdot \frac{N^n}{\log n}.$$

Proof. We apply Theorems 45, 48 and 46. □

3.1 Partial functions

One does not always look for realizing polynomials for fully defined functions. There are many situations, when one only needs a realizing polynomial which fulfills several criteria, e.g. attains predefined values at only certain inputs, not on all inputs. When a function is not necessarily given on the whole domain, we call it a *partial function*. We already used this notion in Chapter 2: in the proof of Theorem 18, more precisely in Lemma 21: the function $f_b^{(2)}$ was not defined for every pair of group elements. $\chi_{a;b}$ was, on the contrary, defined for every group element input. This did not cause any confusion in Chapter 2 as we always made clear exactly what we were looking for. From now on, by *'function'* we always mean *possibly partial function,* and we always determine the exact domain at where we require predetermined values of the function. Moreover, we are looking for realizing polynomials not only for functions, but for partial functions, too.

In this Section we make some easy observations about the connection of partial functions over different functionally complete algebras. More precisely, if one functionally complete algebra contains another, then every function over the smaller algebra can be realized shorter or equally long over the larger algebra. For this to make sense, we have to define the length of a partial function.

Definition 53. Let f be an n-ary partial function over an algebra $\mathbf{A}$. Let the domain of f be $D \subseteq A^n$. Then let us denote the length of f with $\|f\|_{\mathbf{A}}$ and define it as:

$$\|f\|_{\mathbf{A}} = \min \{ \|p\|_{\mathbf{A}} : p \text{ polynomial realizes } f \text{ on the domain } D \}.$$

Similarly we define the number of variable occurrences $v_{\mathbf{A}}(f)$:

$$v_{\mathbf{A}}(f) = \min \{ v_{\mathbf{A}}(p) : p \text{ polynomial realizes } f \text{ on the domain } D \}.$$

These definitions agree with the definitions for the case, when $D = A^n$. We note that for a partial function f there does not necessarily exist a polynomial p over the algebra $\mathbf{A}$ such that $\|f\|_{\mathbf{A}} = \|p\|_{\mathbf{A}}$ and $v_{\mathbf{A}}(f) = v_{\mathbf{A}}(p)$. We show such an example in Remark 81.

The following proposition makes some connection between length of functions and partial functions:

Proposition 54. *Let* $\mathbf{A}$ *be a functionally complete algebra and* g *be an* n*-ary partial function on domain* $D \subseteq A^n$*. Then its length [number of variable*

occurrences] is the minimum length [number of variable occurrences] of functions f with domain A^n agreeing with g on D:

$$\begin{aligned} v(g) &= \min_{f|_D=g} v(f), \\ \|g\| &= \min_{f|_D=g} \|f\|. \end{aligned}$$

Proof. In the proof we denote polynomials by p and functions with domain A^n by f. By the definition we have

$$\begin{aligned} v(g) &= \min_p \{ v_{\mathbf{A}}(p) : p \text{ realizes } g \text{ on the domain } D \} = \\ &= \min_p \min_{f|_D=g} \{ v_{\mathbf{A}}(p) : p \text{ realizes } f \} = \\ &= \min_{f|_D=g} \min_p \{ v_{\mathbf{A}}(p) : p \text{ realizes } f \} = \\ &= \min_{f|_D=g} v(f). \end{aligned}$$

The very same argument works for the length $\|g\|$. □

Now let us make some observation about partial functions over different functionally complete algebras.

Proposition 55. *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two functionally complete algebras with the same signature. Let us suppose that there exists an embedding $e\colon \mathbf{A}_1 \hookrightarrow \mathbf{A}_2$. Let $e^n\colon \mathbf{A}_1^n \hookrightarrow \mathbf{A}_2^n$ be the nth power of the embedding e. Let us denote with $(e^n)^{-1}$ the partial inverse of e^n. Let $f\colon A_1^n \to A_1$ be a (possibly partial) function.*

$$\begin{array}{ccc} \mathbf{A}_1^n & \xrightarrow{\quad f \quad} & \mathbf{A}_1 \\ \Big\downarrow{\scriptstyle e^n} & & \Big\downarrow{\scriptstyle e} \\ \mathbf{A}_2^n & \xrightarrow[e\circ f\circ (e^n)^{-1}]{} & \mathbf{A}_2 \end{array}$$

Then

$$\begin{aligned} \left\| e \circ f \circ (e^n)^{-1} \right\|_{\mathbf{A}_2} &\leq \|f\|_{\mathbf{A}_1}, \\ v_{\mathbf{A}_2}\left(e \circ f \circ (e^n)^{-1} \right) &\leq v_{\mathbf{A}_1}(f). \end{aligned}$$

Proof. First we note that $e \circ f \circ (e^n)^{-1}$ is a well defined partial function since e^n is an embedding. The domain of $e \circ f \circ (e^n)^{-1}$ is the image of $\mathbf{A}_1^n$ under e^n. Let p be a polynomial over $\mathbf{A}_1$ which realizes f with $\|p\|_{\mathbf{A}_1} = \|f\|_{\mathbf{A}_1}$. Now,

p is a polynomial over $\mathbf{A}_2$, too and realizes the (possibly partial) function $e \circ f \circ (e^n)^{-1}$, hence $\left\| e \circ f \circ (e^n)^{-1} \right\|_{\mathbf{A}_2} \leq \|p\|_{\mathbf{A}_2} = \|p\|_{\mathbf{A}_1} = \|f\|_{\mathbf{A}_1}$. Similarly, let q be a polynomial over $\mathbf{A}_1$ which realizes f with $v_{\mathbf{A}_1}(q) = v_{\mathbf{A}_1}(f)$. Now, q is a polynomial over $\mathbf{A}_2$, too and realizes the (possibly partial) function $e \circ f \circ (e^n)^{-1}$, hence $v_{\mathbf{A}_2}\left(e \circ f \circ (e^n)^{-1}\right) \leq v_{\mathbf{A}_2}(q) = v_{\mathbf{A}_1}(q) = v_{\mathbf{A}_1}(f)$. □

Corollary 56. *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two functionally complete algebras. Let us suppose that $\mathbf{A}_1 \leq \mathbf{A}_2$. Let f be a (possibly partial) function over $\mathbf{A}_1$ (so it is a possibly partial function over $\mathbf{A}_2$, too). Then*

$$\begin{aligned} \|f\|_{\mathbf{A}_2} &\leq \|f\|_{\mathbf{A}_1}, \\ v_{\mathbf{A}_2}(f) &\leq v_{\mathbf{A}_1}(f). \end{aligned}$$

This proposition and corollary basically tell us that the 'larger' the algebra, the shorter the possible realizing polynomials are. Therefore in the later Sections of the Chapter we do not necessarily search for realizing polynomials over every functionally complete algebra, but only over those which contain the others. This property is especially useful among finite groups as we explain it in Section 3.4.

We determined some upper bounds for several functions over different functionally complete algebras in Theorem 45 and in Theorem 48. Even these theorems can be applied to partial functions, as we just consider them as functions which take value zero where they were not defined originally. If, however, the partial function is only defined on a domain which is subset of $S_1 \times \cdots \times S_n$ for some subsets $S_i \subseteq A$, then we can benefit more. For this we need the notion of the partial characteristic function over the domain $S_1 \times \cdots \times S_n$: let $\chi_{a_1,\ldots,a_n}^{S_1 \times \cdots \times S_n}(x_1, \ldots, x_n)$ be the n-ary partial function for which $\chi_{a_1,\ldots,a_n}^{S_1 \times \cdots \times S_n}(a_1, \ldots, a_n) = 1$ and $\chi_{a_1,\ldots,a_n}^{S_1 \times \cdots \times S_n}(x_1, \ldots, x_n) = 0$ if every $x_i \in S_i$ and for some i_0 we have $x_{i_0} \neq a_{i_0}$. In this definition we require that every $a_i \in S_i$. The domain of $\chi_{a_1,\ldots,a_n}^{S_1 \times \cdots \times S_n}(x_1, \ldots, x_n)$ is $S_1 \times \cdots \times S_n$.

Theorem 57. *Let $\mathbf{A}$ be a functionally complete algebra. Let $0, 1 \in A$ be two distinct elements and let $+, \cdot$ be shortest polynomials with properties such as in Theorem 6. Let χ_a^S be a shortest polynomial representing the unary partial characteristic function for the element a on domain S. Let $\chi_{a_1,\ldots,a_n}^{S_1 \times \cdots \times S_n}$ be a shortest polynomial realizing the n-ary partial characteristic function for the n-tuple $(a_1, \ldots, a_n)$ on domain $S_1 \times \cdots \times S_n$. Let $-$ be a shortest polynomial such that $x - y = 0$ if and only if $x = y$ and $1 - 0 = 1$. Let p be a shortest polynomial realizing an arbitrary n-ary partial function f over $\mathbf{A}$ with e-many non-zero values ($1 \leq e \leq \prod_{i=1}^{n} |S_i|$) on domain $S_1 \times \cdots \times S_n$. For a set S let*

$S - a = \{ s - a : s \in S \}$. *Then the following inequalities hold:*

$$\begin{aligned}
&\left\|\chi_a^S\right\| \leq \|-\| \cdot \left\|\chi_0^{S-a}\right\|, \\
&\left\|\chi_{a_1,\ldots,a_n}^{S_1\times\cdots\times S_n}\right\| \leq 2 \cdot \|\cdot\| \cdot n^{\log v(\cdot)} \cdot \max\left\{ \left\|\chi_{a_i}^{S_i}\right\| : 1 \leq i \leq n \right\}, \\
&\|p\| \leq 2 \cdot \|\cdot\| \cdot \|+\| \cdot e^{\log v(+)} \cdot \max\left\{ \left\|\chi_{a_1,\ldots,a_n}^{S_1\times\cdots\times S_n}\right\| : a_i \in A \right\}, \\
&\|p\| \leq 4 \cdot \|-\| \cdot \|\cdot\|^2 \cdot \|+\| \cdot e^{\log v(+)} \cdot n^{\log v(\cdot)} \cdot \max\left\{ \left\|\chi_0^{S_i-a}\right\| : 1 \leq i \leq n \right\}.
\end{aligned}$$

Proof. The proof is simply applying Lemma 39 and Lemma 44 on the following representations:

$$\begin{aligned}
\chi_a^S(x) &= \chi_0^{S-a}(x-a), \\
\chi_{a_1,\ldots,a_n}^{S_1\times\cdots\times S_n}(x_1,\ldots,x_n) &= \prod_{i=1}^{n} \chi_{a_i}^{S_i}(x_i), \\
p(x_1,\ldots,x_n) &= \sum_{(a_1,\ldots,a_n)\in A^n} \left(p(a_1,\ldots,a_n) \cdot \chi_{a_1,\ldots,a_n}^{S_1\times\cdots\times S_n}(x_1,\ldots,x_n) \right).
\end{aligned}$$

Here we consider $\prod$ and $\sum$ as the iterated versions of $\cdot$ and $+$ in the way described in Lemma 44. □

As we see, there is not too much to gain: we might be able to shorten our polynomials if we can represent χ_a^S shorter than χ_a. We note here that Theorem 48 has a 'partial' version, too. The proof goes exactly as the proof of Theorem 48, so we only state the theorem here:

Theorem 58. *Let* $\mathbf{A}$ *be a functionally complete algebra. Let* $0, 1 \in A$ *be two distinct elements and let* $+$, $\cdot$ *be shortest polynomials with properties such as in Theorem 6 and let us suppose that* $\|+\| = v(+) = v(\cdot) = \|\cdot\| = 2$. *Let* χ_a^S *be a shortest polynomial representing the unary partial characteristic function for the element* a *on domain* S. *Let* $\chi_{a_1,\ldots,a_n}^{S_1\times\cdots\times S_n}$ *be a shortest polynomial realizing the* n*-ary partial characteristic function for the* n*-tuple* $(a_1,\ldots,a_n)$ *on domain* $S_1 \times \cdots \times S_n$. *Let* $-$ *be a shortest polynomial such that* $x - y = 0$ *if and only if* $x = y$ *and* $1 - 0 = 1$. *Let* p *be a shortest polynomial realizing an arbitrary* n*-ary partial function* f *over* $\mathbf{A}$ *with* e*-many non-zero values* $(1 \leq e \leq \prod_{i=1}^{n} |S_i|)$ *on domain* $S_1 \times \cdots \times S_n$. *Let* $s = \max\{ |S_i| : 1 \leq i \leq n \}$ *and let* T *be any positive real number for which* $T \geq \max\left\{ \left\|\chi_a^S\right\| : a \in S, S = S_i, 1 \leq i \leq n \right\}$. *Then the following inequalities*

hold if $s \geq 3$*:*

$$\|p\| \leq e \cdot (1 + T \cdot (3 + n - \log_s e)) - 2 \cdot T,$$
$$\|p\| \leq e \cdot \left(1 + T \cdot \left(3 + n - \frac{\log e}{\log s}\right)\right) - 2 \cdot T,$$

If $s = 2$*, then*

$$\|p\| \leq ((3 + n - \log e) \cdot e - 2) \cdot T.$$

Again, we see that basically $|A|$ is changed to s, the maximum of the number of elements in one coordinate of the domain set and T might be decreased depending on the algebra. We do not give analogous theorems in the later Sections, as the proofs are similar: they just use unary partial characteristic functions on a subset, rather than on the whole algebra. It is still interesting to know what algebras can benefit from considering only partial functions on a domain $S_1 \times \cdots \times S_n$, so we always make a note for particular algebras in the remaining part of this Chapter. This property can be beneficial if a functionally complete algebra is embedded into another one and we want to realize a function of the smaller algebra over the larger one.

3.2 The two-element Boolean algebra

First we consider the two-element functionally complete algebras, especially $\mathbf{B}$ and $\mathbf{B}_0$. Let us start with the observation that over $\mathbf{B}$ we only have to use negation in front of variables:

Proposition 59. *Let* f *be a (possibly partial) function over* $\{0, 1\}$*. Then there exist two polynomials* p_1, p_2 *realizing* f *such that every negation in* p_1 *and* p_2 *is only used on variables, moreover* $\|f\| = \|p_1\|$ *and* $v(f) = v(p_2)$*.*

Proof. The proof is a basic one in mathematical logic, thus we just sketch it. One can find more details, in e.g. [6]. We define the *level of a polynomial*. A constant or a variable has level 0 and if p_1 and p_2 are two polynomials over $\mathbf{B}$ with level l_1 and l_2, then the level of $(p_1 \wedge p_2)$ is $1 + \max(l_1, l_2)$, the level of $(p_1 \vee p_2)$ is $1 + \max(l_1, l_2)$ and the level of $\neg p_1$ is $1 + l_1$.

Let us observe that $\neg(x \wedge y) = \neg x \vee \neg y$, $\neg(x \vee y) = \neg x \wedge \neg y$ and $\neg\neg x = x$. Now let p_1 be a polynomial which represents f and for which $\|p_1\| = \|f\|$. If p_1 contains any negation which is not a negation of a variable, then it either negates a negation, a $\wedge$ or a $\vee$. Let us substitute this negation using the corresponding above-mentioned rule, this does not change the length of

the polynomial. Let L be the ordered list of polynomials q which appear negated (i.e. as $\neg q$) in p_1. After a substitution of a negation in p_1 using any of the above-mentioned rules, a polynomial from L is removed and some new polynomials are added. Each of the new added polynomials have strictly less level than the removed polynomial.

If e.g. p_1 has a subpolynomial $\neg(x \wedge y)$, then $x \wedge y$ appears in L. When we substitute every appearance of $\neg(x \wedge y)$ to $\neg x \vee \neg y$, then the polynomial $x \wedge y$ is removed from L and x and y is added to it. The polynomial $x \wedge y$ has level 1, the variables x and y have level 0.

Iterating this algorithm ends in a polynomial, when L only contains variables, i.e. every negation negates a variable or a constant. Replacing the negation of the constants by the appropriate corresponding constants finishes the proof.

Using the same idea we have the result for the number of variable occurrences. □

In the following proposition we compare the length and the number of variable occurrences for $\mathbf{B}$ and for $\mathbf{B}_0$.

Proposition 60. *Let f be an n-ary (possibly partial) function over $\{0,1\}$. Then*

$$\begin{aligned} v_{\mathbf{B}}(f) &= v_{\mathbf{B}_0}(f) \\ \|f\|_{\mathbf{B}} &\leq \|f\|_{\mathbf{B}_0} \leq 3 \cdot \|f\|_{\mathbf{B}} \end{aligned}$$

Proof. The proof based on an easy observation, namely that NAND and NOR only differ from $\wedge$ and $\vee$ by a negation. Now let p be a polynomial over $\mathbf{B}$, which realizes f. By Proposition 59 we can assume that every negation in p negates a variable. Now changing every $\wedge, \vee, \neg$ using the rules (2.2), (2.3) and (2.4) we do not change the number of variable occurrences, but we increase the length by 1 each time. As p had negations only in front of variables we can conclude that $v_{\mathbf{B}_0}(f) \leq v_{\mathbf{B}}(f)$ and $\|f\|_{\mathbf{B}_0} \leq 3 \cdot \|f\|_{\mathbf{B}}$.

Now if p realizes f over $\mathbf{B}_0$, then substituting for NAND and NOR using x NAND $y = \neg(x \wedge y)$ and x NOR $y = \neg(x \vee y)$ we increase neither the length nor the number of variable occurrences of the polynomial. Hence, we conclude that $v_{\mathbf{B}}(f) \leq v_{\mathbf{B}_0}(f)$ and $\|f\|_{\mathbf{B}} \leq \|f\|_{\mathbf{B}_0}$. □

Later on we only consider realizing polynomials over $\mathbf{B}$. One can give estimations on the length and on the number of variable occurrences over $\mathbf{B}_0$ using Proposition 60.

Now we give some upper bounds on the length for an arbitrary function over $\mathbf{B}$. We basically use the idea of Theorem 48.

Theorem 61. *Let* $\mathbf{B}$ *be the two-element Boolean algebra. Let* f *be an arbitrary* n*-ary function over* $\{0,1\}$ *with* e*-many non-zero values* ($1 \leq e \leq 2^n$). *Then*

$$\begin{aligned} v_{\mathbf{B}}(f) &= \|f\|_{\mathbf{B}}, \\ \|f\|_{\mathbf{B}} &\leq (3 + n - \log e) \cdot e - 2. \end{aligned}$$

Proof. Let p be a realizing polynomial for f with minimal length among those polynomials which have exactly $v(f)$-many variable occurrences. If any constants appear in p, then first let us change every negation of a constant using the rules $\neg 0 = 1$ and $\neg 1 = 0$. Now every appearance of a constant has one of the following forms: $0 \wedge p'$, $1 \wedge p'$, $0 \vee p'$, $1 \vee p'$ for some subpolynomial p'. Since $0 \wedge p' = 0$, $1 \wedge p' = 0 \vee p' = p'$, $1 \vee p' = 1$, we could shorten p if any of these forms appear. Therefore we have $\|f\| = v(f)$.

The inequality for the length follows from Theorem 48 ($\vee$ plays the role of the addition, $\wedge$ is the multiplication) and from the fact that both unary characteristic functions $\chi_0(x) = \neg x$, $\chi_1(x) = x$ have length 1. □

Remark 62. We can assume that $e \leq 2^{n-1}$, otherwise we realize $\neg f$ with p, then $\neg p$ realizes f and has the same length as p.

Remark 63. Lupanov [25] considered the algebra over $\{0,1\}$ which contains all 16 binary operations as basic operations. He proved that an arbitrary n-ary function can be realized with length at most $(2 + o(1)) \cdot 2^n \cdot (\log n)^{-1}$ over this algebra. Our bound is better than Lupanov's, whenever $e < c \cdot 2^n \cdot (\log n)^{-2}$ for some constant c.

The following Corollary summarizes our upper and lower bounds for the algebras $\mathbf{B}$ and $\mathbf{B}_0$:

Corollary 64. *Let* f *be an arbitrary* n*-ary function over* $\{0,1\}$. *Then*

$$\begin{aligned} \|f\|_{\mathbf{B}} &\leq 2 \cdot 2^n - 2, \\ \|f\|_{\mathbf{B}_0} &\leq 6 \cdot 2^n - 6. \end{aligned}$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_0 *over* $\{0,1\}$ *such that*

$$\begin{aligned} \|f_0\|_{\mathbf{B}} &\geq \frac{1}{1+\varepsilon} \cdot \frac{2^n}{\log n}, \\ \|f_0\|_{\mathbf{B}_0} &\geq \frac{1}{1+\varepsilon} \cdot \frac{2^n}{\log n}. \end{aligned}$$

Proof. We apply Proposition 60, Theorem 61, Remark 62 and Theorem 46. □

The last proposition of this Section gives an upper bound on the length of the discriminator operation.

Proposition 65. *Let d be the discriminator function over $\{0,1\}$. Then*

$$\|d\|_{\mathbf{B}} \leq 10.$$

Proof. In the second proof of Theorem 14 we gave a polynomial which has length 10 and realizes the discriminator function. □

As the only functionally complete Boolean algebra has 2 elements, there is no point considering partial functions over the set $S_1 \times \cdots \times S_n$. If k is the number of S_i's for which $|S_i| = 2$, then we can easily consider a function over $\mathbf{B}^k$ instead of the original partial function.

3.3 Finite rings

So far we did not define exactly what are the basic operations of a ring, but as soon as we are considering the length of polynomials, we have to be exact. From now on the rings basic operations are the $+$, $-$ and $\cdot$. Let $\mathbf{F}$ be a finite field, let $q = |\mathbf{F}|$ and let $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$, the $k \times k$-matrices over $\mathbf{F}$. Now $\|+\| = v(+) = \|\cdot\| = v(\cdot) = \|-\| = v(-) = 2$, hence we are able to apply Theorem 48. It is easy to see that the n-ary addition and multiplication both have length n. First we start with the finite fields.

Theorem 66. *Let $\mathbf{F}$ be a finite field, $|\mathbf{F}| = q$ and let f be an arbitrary n-ary function over $\mathbf{F}$ with e-many non-zero values ($1 \leq e \leq q^n$). Then*

$$\|f\| \leq \left((2q-2) \cdot \left(3 + n - \log_q e\right) + 1\right) \cdot e - 4q + 4,$$
$$\|f\| \leq 2 \cdot q \cdot e \cdot \left(3 + n - \log_q e\right).$$

Proof. We have $\chi_0(x) = 1 - x^{q-1}$, which has length $q \leq 2q-2$. For every $a \neq 0$ let $c_a = \left(\prod_{a \neq u \in \mathbf{F}} (a-u)\right)^{-1}$. Then $\chi_a(x) = c_a \cdot \prod_{a \neq u \in \mathbf{F}} (x-u)$, hence $\|\chi_a\| \leq 2q-2$ (as one of the u's is zero). Now applying Theorem 48 we obtain the required bounds. □

Remark 67. We note that if a partial function is defined over a domain $S_1 \times \cdots \times S_n$, and $s = \max\{|S_i| : 1 \leq i \leq n\}$, then Theorem 66 holds if we change q to s, i.e. $\|f\| \leq ((2s-2) \cdot (3 + n - \log_s e) + 1) \cdot e - 4s + 4$.

Now we move to the $k \times k$ matrix rings. Let $N = q^{k^2}$ the number of elements of the $k \times k$ matrix ring over the q-element field. The following theorem gives us an upper bound on the length of an arbitrary n-ary function.

Theorem 68. *Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$ *and let* $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$*, the* $k \times k$*-matrices over* $\mathbf{F}$ *(*$k \geq 2$*). Let* $N = |\mathbf{M}_k(\mathbf{F})| = q^{k^2}$ *and let* f *be an arbitrary* n*-ary function over* $\mathbf{R}$ *with* e*-many non-zero values (*$1 \leq e \leq N^n$*). Then*

$$\|f\| \leq 16 \cdot (\log N)^{5/2} \cdot N^{1/4} \cdot e \cdot (3 + n - \log_N e)\,.$$

Proof. We use the notations of the proof of Theorem 16. We recall that $X \vee Y = X + Y - X \cdot Y$. Let us define the polynomial $\bigvee_{i=1}^{n} X_i$ the way we described in Lemma 44. Let $v(n)$ be the number of variable occurrences in $\bigvee_{i=1}^{n}$. $\vee$ has 4 variable occurrences, hence $v(n) \leq 4 \cdot n^2$ by Lemma 44. Moreover for expressing $\bigvee_{i=1}^{n}$ we do not need to have any constants. Thus for every n we have $\|\bigvee_{i=1}^{n}\| \leq 4n^2$.

Let us recall the following polynomials from the proof of Theorem 16:

$$\begin{aligned} p_{i,j}(X) &= \sum_{s=1}^{k} I_{s,i} \cdot X \cdot I_{j,s}, \\ \delta(X) &= \bigvee_{i,j=1}^{k} \left(p_{i,j}(X)^{q-1}\right), \\ \chi_M(X) &= I - \delta(X - M) \end{aligned}$$

It is easy to see that $\|p_{i,j}\| = 3k$, $v(p_{i,j}) = k$. Now, applying Lemma 40 and Lemma 41 we have

$$\begin{aligned} v(\delta) &\leq v\left(k^2\right) \cdot v(p_{i,j}) \cdot (q-1) \leq 4 \cdot k^5 \cdot (q-1)\,, \\ \|\delta\| &\leq v\left(k^2\right) \cdot (q-1) \cdot \|p_{i,j}\| \leq 12 \cdot k^5 \cdot (q-1)\,, \\ \|\chi_M\| &\leq \|\delta\| + v(\delta) \cdot (\|-\| - 1) + 1 \leq 16 \cdot k^5 \cdot (q-1) + 1. \end{aligned}$$

Let us denote with T the right hand side of the last inequality and apply Theorem 48. Then we derive the following bound on $\|f\|$:

$$\|f\| \leq e \cdot \left(1 + \left(16 \cdot k^5 \cdot (q-1) + 1\right) \cdot \left(3 + n - \frac{\log e}{k^2 \cdot \log q}\right)\right) - 32 \cdot k^5 \cdot (q-1) - 1.$$

Now using $k^5 = \left(\log_q N\right)^{5/2} \leq (\log N)^{5/2}$ and $q = N^{1/q^2} \leq N^{1/4}$ we easily derive the desired bound. □

Remark 69. If $e > (N-1) \cdot N^{n-1}$, then there exists a value $0 \neq r \in \mathbf{R}$ such that f takes the value r at least N^{n-1}-many times. Let us realize $f - r$ with p, then $p + r$ realizes f, $v(p + r) = v(p)$ and $\|p + r\| \leq \|p\| + 1$. Therefore we can assume that $e \leq (N-1) \cdot N^{n-1}$.

Remark 70. Building up the characteristic function over finite matrix rings is somewhat different than building them up over finite fields. Over fields the polynomial for $\chi_a(x)$ checks whether the input is different than any element $u \in \mathbf{F}$ (apart from $u = a$). On the other hand, in the case of matrix rings, the polynomial checks whether all the *entries* of the matrix differ from anything but zero. Hence if we want to have a theorem about partial functions, we have to make a restrictions on the entries of the domain, like the entry in the ith row and jth column has to be from the set $S_{i,j} \subseteq \mathbf{F}$. Then Theorem 68 applies, if we change N to $\prod_{i,j=1}^{n} |S_{i,j}|$.

Let us summarize our upper and lower bounds for functionally complete rings:

Corollary 71. *Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$. *For an arbitrary* n*-ary function* f *over* $\mathbf{F}$ *we have*

$$\|f\|_{\mathbf{F}} \leq 10 \cdot (q-1)^2 \cdot q^{n-1}.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_0 *over* $\mathbf{F}$ *such that*

$$\|f_0\|_{\mathbf{F}} \geq \frac{\log q}{1+\varepsilon} \cdot \frac{q^n}{\log n}.$$

Let $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$, *the* $k \times k$ *matrices over* $\mathbf{F}$ *(*$k \geq 2$*) and let* $N = |\mathbf{M}_k(\mathbf{F})| = q^{k^2}$. *For an arbitrary* n*-ary function* f' *over* $\mathbf{F}$ *we have*

$$\|f'\|_{\mathbf{R}} \leq 80 \cdot (\log N)^{5/2} \cdot (N-1) \cdot N^{n-1+1/4}.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_1 *over* $\mathbf{R}$ *such that*

$$\|f_1\|_{\mathbf{R}} \geq \frac{\log N}{1+\varepsilon} \cdot \frac{N^n}{\log n}.$$

Proof. We apply Theorems 66, 68, Remark 69 and Theorem 46. □

We finish the Section with the upper bounds on the discriminator function. The following propositions show that it is linear in the size of the ring $\mathbf{R}$.

Proposition 72. *Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$ *and let* d *be the discriminator function over* $\mathbf{F}$. *Then*

$$\|d\| \leq 4 \cdot q - 1.$$

Proof. As we showed in the second proof of Theorem 16, the discriminator has the following polynomial realization:

$$d(x,y,z) = (x-y)^{q-1} \cdot x + \left(1-(x-y)^{q-1}\right) \cdot z.$$

It is easy to see that this polynomial has length $4 \cdot q - 1$. □

Proposition 73. *Let* $\mathbf{F}$ *be a finite field,* $|\mathbf{F}| = q$ *and let* $\mathbf{R} = \mathbf{M}_k(\mathbf{F})$*, the* $k \times k$*-matrices over* $\mathbf{F}$ *(*$k \geq 2$*). Let* $N = |\mathbf{M}_k(\mathbf{F})| = q^{k^2}$ *and let* d *be the discriminator function over* $\mathbf{R}$*. Then*

$$\|d\| \leq 32 \cdot N^{1/4} \cdot (\log N)^{5/2}.$$

Proof. As we showed it in the second proof of Theorem 16, the discriminator has the following polynomial realization:

$$d(X,Y,Z) = \delta(X-Y) \cdot X + (I - \delta(X-Y)) \cdot Z.$$

Using Lemma 41 we can give an upper bound on the length:

$$\|d\| \leq 2 \cdot (\|\delta\| + v(\delta)) + 3 \leq 32 \cdot k^5 \cdot (q-1) + 3 \leq 32 \cdot q \cdot k^5.$$

As in the proof of Theorem 68 we proved that if $q \leq N^{1/4}$ and $k^5 \leq (\log N)^{5/2}$. □

We do not claim that a shortest polynomial p realizing an arbitrary function f is necessarily built up the way we obtained the bounds in this Section. An interesting question is to find the minimal length of a realizing polynomial for an arbitrary (or special) function and whether it can be found in a fast way. Another interesting question is whether the shortest realizing polynomial is unique for every function f, and if not, then characterize those functions for which the shortest realizing polynomial is unique.

3.4 Finite groups

So far we gave bounds on the length of arbitrary n-ary functions for the two-element Boolean algebra and for the functionally complete rings. Every upper bound was based on Theorem 48 and on the idea that we build up our given function recursively. Theorem 48, however, uses some strict conditions, namely that some $+$ and $\cdot$ operations must have exactly two variable occurrences. Among groups $+$ naturally corresponds to the usual multiplication of the group, but there is no short or natural function corresponding to $\cdot$. Unfortunately, as we stated in Remark 50, it is essential that both $v(+) = 2$

and $v(\cdot) = 2$. Therefore we have to find another way for giving bounds on the length and on the number of variable occurrences for an arbitrary n-ary function over a functionally complete group. We use the idea of Theorem 45 and the proof of Theorem 18 helps us to build up our polynomials.

Throughout this Section let $\mathbf{G}$ be a finite simple non-Abelian group with two basic operations: the group multiplication and the inverse. Let $N = |\mathbf{G}|$. We write $[x, y]$ for the commutator $x^{-1}y^{-1}xy$ and put $x^y = y^{-1}xy$. First we observe that the variable number of occurrences is connected to the length of a function and the realizing polynomials can be chosen such that all inverses are taken only on variables.

Proposition 74. *Let f be an arbitrary n-ary (possibly partial) function over $\mathbf{G}$. Then there exists realizing polynomials p_1 and p_2, such that every inverse is used only on variables and $\|f\| = \|p_1\|$, $v(f) = v(p_2)$. Moreover,*

$$\|f\|_{\mathbf{G}} \leq 2 \cdot v_{\mathbf{G}}(f) + 1.$$

Proof. Proving that it is enough to consider polynomials with only variables inverted is entirely the same as the proof of Proposition 59 for the two-element Boolean algebra. We iterate substituting every invers of a product $(xy)^{-1}$ by $y^{-1}x^{-1}$. This operations changes neither the length, nor the number of variable occurrences of the polynomial. When the algorithm terminates, the resulting polynomial will have the required property.

For proving the inequality let p be a polynomial over $\mathbf{G}$ which realizes f and $v(p) = v(f)$. Then there exists a polynomial p' which realizes f, $v(p') = v(p)$ and $\|p'\| \leq 2v(p)+1$: we replace in p every product of constants $c_1 \cdots c_k$ by the constant c, where $c = c_1 \cdots c_k$. Then $v(p') = v(p)$ and there must be at least 1 variable between every two constants, hence $\|p'\| \leq 2 \cdot v(p') + 1$. Now

$$\|f\| \leq \|p'\| \leq 2 \cdot v(p') + 1 = 2 \cdot v(p) + 1 = 2 \cdot v(f) + 1.$$

□

From now on we only consider the number of variable occurrences of a function, and one can derive a bound for the length using Proposition 74. We use Lemma 40 for estimating the number of variable occurrences in the (partial) functions given in the proof of Theorem 18. We remind the reader for some notations defined in the proof of Theorem 18.

For every $1 \neq u \in G$ and for every $v \in G$ let $p_{u,v}$ be the unary partial function for which $p_{u,v}(1) = 1$ and $p_{u,v}(u) = v$. Let $f_b^{(n)}$ (for $b \neq 1$) be the n-ary partial function defined in Lemma 21, i.e. $f_b^{(n)}(b, \ldots, b) = b$ and $f_b^{(n)}(x_1, \ldots, x_n) = 1$ if $x_i = 1$ for some $1 \leq i \leq n$. Let $\chi_{1;u}$ (for

$u \neq 1$) be the unary characteristic function described in Lemma 23, i.e. $\chi_{1;u}(1) = u$ and $\chi_{1;u}(x) = 1$ if $x \neq 1$. Finally let $\chi_{a_1,\ldots,a_n;u}$ be the n-ary characteristic function described in Lemma 25, i.e. $\chi_{a_1,\ldots,a_n;u}(a_1,\ldots,a_n) = u$ and $\chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n) = 1$, whenever $x_i \neq a_i$ for some i.

Let $V = v\left(f_b^{(2)}\right)$. For every $1 \neq u \in G$, for every $v \in G$, and for every subset $S \subseteq G$ let

$$\begin{aligned} K_{u,v} &= v(p_{u,v}), \\ K_{S,v} &= \max\{ K_{u,v} : 1 \neq u \in S \}, \\ K_{u,S} &= \max\{ K_{u,v} : v \in S \}. \end{aligned}$$

Later on we usually use u to denote an arbitrary element of $G \setminus \{1\}$, use v as an arbitrary element of G, and use b whenever we are referring to a (somehow) *fixed* element of $G \setminus \{1\}$.

Theorem 75. *Let* $\mathbf{G}$ *be a functionally complete group. Let* $N = |\mathbf{G}|$. *Then the following inequalities hold:*

$$v\left(f_b^{(n)}\right) \leq V \cdot n^{\log V}, \tag{3.16}$$

$$v(\chi_{1;b}) \leq v\left(f_b^{(N-1)}\right) \cdot \max\{ v(p_{u,b}) : 1 \neq u \in G \}, \tag{3.17}$$

$$v(\chi_{1;u}) \leq v(\chi_{1;b}) \cdot v(p_{b,u}), \tag{3.18}$$

$$v(\chi_{a_1,\ldots,a_n;b}) \leq v\left(f_b^{(n)}\right) \cdot v(\chi_{1;b}), \tag{3.19}$$

$$v(\chi_{a_1,\ldots,a_n;u}) \leq v(\chi_{a_1,\ldots,a_n;b}) \cdot v(p_{b,u}), \tag{3.20}$$

Let f *be an* n*-ary (possibly partial) function over* $\mathbf{G}$ *with* e*-many non-identity values* ($1 \leq e \leq N^n$). *Let* $K = 1 + \max\left\{ K_{G\setminus\{1\},b}, K_{b,G\setminus\{1\}} \right\}$. *Then* K *is at most the number of conjugacy classes of* $\mathbf{G}$ *and*

$$v(f) \leq e \cdot \max\{ v(\chi_{a_1,\ldots,a_n;u}) : 1 \neq u \in G \}, \tag{3.21}$$

$$v(f) \leq e \cdot v\left(f_b^{(n)}\right) \cdot v\left(f_b^{(N-1)}\right) \cdot \max_{1 \neq u_1 \in G} v(p_{u_1,b}) \cdot \max_{1 \neq u_2 \in \mathbf{G}} v(p_{b,u_2}), \tag{3.22}$$

$$v(f) \leq e \cdot K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}} \cdot V^2 \cdot n^{\log V} \cdot (N-1)^{\log V}, \tag{3.23}$$

$$v(f) \leq 3136 \cdot (K-1)^2 \cdot (N-1)^8 \cdot n^8 \cdot e, \tag{3.24}$$

$$\|f\| \leq 2 \cdot K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}} \cdot V^2 \cdot (N-1)^{\log V} \cdot n^{\log V} \cdot e + 1, \tag{3.25}$$

$$\|f\| \leq 6272 \cdot (K-1)^2 \cdot (N-1)^8 \cdot n^8 \cdot e + 1. \tag{3.26}$$

Proof. For proving (3.16) we use Lemma 44 on the polynomials $p^{(n)}$, where $p^{(1)}(x_1) = x_1$, $p^{(2)}(x_1, x_2)$ is a realizing polynomial for $f_b^{(2)}$ such that $v\left(p^{(2)}\right) =$

V, and for every integer $n \geq 2$:

$$\begin{aligned} p^{(2n-1)}(x_1,\ldots,x_{2n-1}) &= p\left(p^{(n)}(x_1,\ldots,x_n), p^{(n-1)}(x_{n+1},\ldots,x_{2n-1})\right) \\ p^{(2n)}(x_1,\ldots,x_{2n}) &= p\left(p^{(n)}(x_1,\ldots,x_n), p^{(n)}(x_{n+1},\ldots,x_{2n})\right). \end{aligned}$$

Then $p^{(n)}$ is a realizing polynomial for $f_b^{(n)}$ and by Lemma 44 we have $v\left(f_b^{(n)}\right) \leq v\left(p^{(n)}\right) \leq V \cdot n^{\log V}$.

The inequalities (3.17), (3.18), (3.19), (3.20), (3.21) follow from Lemma 40 and the following representations based on the proof of Theorem 18, where $b \neq 1$ and $u \neq 1$:

$$\begin{aligned} \chi_{1;b}(x) &= f_b^{(N-1)}\left(bp_{u_2,b}(x)^{-1},\ldots,bp_{u_N,b}(x)^{-1}\right), \\ \chi_{1;u}(x) &= p_{b,u}\left(\chi_{1;b}(x)\right), \\ \chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n) &= f_b^{(n)}\left(\chi_{1;b}\left(x_1a_1^{-1}\right),\ldots,\chi_{1;b}\left(x_na_n^{-1}\right)\right), \\ \chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n) &= p_{b,u}\left(\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n)\right), \\ f(x_1\ldots,x_n) &= \prod_{\substack{(a_1,\ldots,a_n)\in G^n \\ 1\neq u=f(a_1\ldots,a_n)}} \chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n), \end{aligned}$$

where $G = \{1, u_2, \ldots, u_N\}$. Then (3.22) and (3.23) simply follows from the first 6 inequalities. The number K is at most the number of conjugacy classes of $\mathbf{G}$ by Proposition 79 (see Section 3.4.1 below). The inequality (3.24) follows if we apply Proposition 86 on (3.23) (see Section 3.4.2 below). Finally, the last two equations are an immediate consequence of the equations (3.23), (3.24) and Proposition 74. □

Remark 76. Similarly to Remarks 62 and 69, if $e > (N-1)\cdot N^{n-1}$, then there exists a value $1 \neq g \in \mathbf{G}$ such that f takes the value g at least N^{n-1}-many times. Let us realize $f \cdot g^{-1}$ with p, then $p \cdot g$ realizes f, $v(p \cdot g) = v(p)$ and $\|p \cdot g\| \leq \|p\| + 1$.

Comparing the results of Theorem 75 with those of Theorem 45 we can conclude that $f_b^{(2)}$ plays some similar role for the groups as the $\cdot$ in general. One wants to minimize V in order to have better upper bounds for $v(f)$, which may be possible to do by choosing b wisely. As we see, e is taken to the first power, as the group multiplication plays the role of the general $+$. The constants $K_{b,G\setminus\{1\}}$ and $K_{G\setminus\{1\},b}$ depend on the choice of b, too. In the following Subsections we give some upper and lower bounds on V and on the $K_{u,v}$'s.

Remark 77. We note here that if $S_1, \ldots, S_n, S \subseteq G$ are subsets, where $1 \in S_1 \cap \cdots \cap S_n$, and f is a partial n-ary function defined over the domain $S_1 \times \cdots \times S_n$ with values from S, then similar inequalities hold as in Theorem 75:

$$
\begin{aligned}
&v\left(\chi_{1;b}^{S_i}\right) \leq v\left(f_b^{(|S_i|-1)}\right) \cdot \max\left\{ v\left(p_{u,b}\right) : 1 \neq u \in S_i \right\},\\
&v\left(\chi_{1;u}^{S_i}\right) \leq v\left(\chi_{1;b}^{S_i}\right) \cdot v\left(p_{b,u}\right),\\
&v\left(\chi_{a_1,\ldots,a_n;b}^{S_1\times\cdots\times S_n}\right) \leq v\left(f_b^{(n)}\right) \cdot \max_{1\leq i\leq n} v\left(\chi_{1;b}^{S_i}\right),\\
&v\left(\chi_{a_1,\ldots,a_n;u}^{S_1\times\cdots\times S_n}\right) \leq v\left(\chi_{a_1,\ldots,a_n;b}^{S_1\times\cdots\times S_n}\right) \cdot v\left(p_{b,u}\right),\\
&v\left(f\right) \leq e \cdot \max\left\{ v\left(\chi_{a_1,\ldots,a_n;u}^{S_1\times\cdots\times S_n}\right) : 1 \neq u \in S \right\},\\
&v\left(f\right) \leq e \cdot K_{b,S\setminus\{1\}} \cdot \max_{1\leq i\leq n} K_{S_i\setminus\{1\},b} \cdot V^2 \cdot n^{\log V} \cdot \max_{1\leq i\leq n} \left(|S_i|-1\right)^{\log V},\\
&v\left(f\right) \leq 3136 \cdot (K-1)^2 \cdot \max_{1\leq i\leq n} \left(|S_i|-1\right)^8 \cdot n^8 \cdot e.
\end{aligned}
$$

The bounds apply even in the slightly weird situation when $|S_i| = 1$ or $|S_i| = 2$. When $|S_i| = 1$ then the corresponding characteristic and f_b^0 functions are constant functions, and have zero variable occurrences. If $|S_i| = 2$ then the corresponding f_b^1 function has one variable occurrence as $f_b^1(x) = x$.

Embedding $\mathbf{G}_1$ into a larger group $\mathbf{G}_2$ may allow us to shorten the length of an arbitrary (partial) function f. Formally we obtain the same upper bounds (as the sets S_i's and S are the same for the two groups), but by the embedding we have a chance to choose b from a larger set. This may enable us to decrease $v\left(f_b^{(2)}\right)$, $v\left(f_b^{(n)}\right)$, and $v\left(p_{u,v}\right)$, hence also to shorten $v(f)$ and $\|f\|$ for the partial function f over $\mathbf{G}_2$.

Let us summarize our bounds for functionally complete groups:

Corollary 78. *Let* $\mathbf{G}$ *be a functionally complete group. Let* $N = |\mathbf{G}|$ *and let* K *be the number of conjugacy classes of* $\mathbf{G}$*. For an arbitrary* n*-ary function* f *over* $\mathbf{G}$ *we have*

$$\|f\|_{\mathbf{G}} \leq 6272 \cdot (K-1)^2 \cdot (N-1)^9 \cdot N^{n-1} \cdot n^8 + 1.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_0 *such that*

$$\|f_0\|_{\mathbf{G}} \geq \frac{\log N}{1+\varepsilon} \cdot \frac{N^n}{\log n}.$$

Proof. We apply Theorem 75, Remark 76 and Theorem 46. □

3.4.1 The partial function $p_{u,v}$

First we give upper bounds on the number of variable occurrences of the partial functions $p_{u,v}$. For the group $\mathbf{G}$ and a set $S \subseteq G$ let

$$S^k = \{ u_1 \cdots u_k \mid u_1, \ldots, u_k \in S \}.$$

For two elements $u, v \in G$ let us denote $u \sim_{\mathbf{G}} v$ if u is a conjugate of v in $\mathbf{G}$. If it is clear over which group we are considering the conjugation, we just write $u \sim v$. Let $C_u = \{ u^c : c \in G \}$ be the conjugacy class of u and let $D_u = C_u \cup C_{u^{-1}}$. We generate every $v \in G$ using the elements of D_u as generators for some $1 \neq u \in G$. Let $S_0 = \emptyset$, $S_1 = D_u$ and for every natural number $i \geq 2$ we will create S_i, a subset of G, using the following definition:

$$S_i = S_{i-1} \cup \{ x \cdot y \mid x \in S_{i-1}, y \in D_u \} = \bigcup_{j=1}^{i} D_u^j.$$

It is clear that $S_i \subseteq S_{i+1}$ and by Lemma 19 it can only terminate in G, i.e. if $S_i = S_{i+1}$, then $S_i = G$. Moreover, S_i is the union of conjugacy classes, hence the process will finish in at most as many steps as the number of the conjugacy classes of $\mathbf{G}$. The following proposition tells us that this is the way to determine the $K_{u,v}$'s.

Proposition 79. *For every $1 \neq u \in G$ and for every $v \in G$ we have $v \in S_i$ if and only if $v\,(p_{u,v}) = K_{u,v} \leq i$. As a corollary we derive that $K_{u,v}$ is always less than the number of conjugacy classes K of $\mathbf{G}$. Moreover,*

$$\|p_{u,v}\| \leq 2 \cdot K_{u,v} + 1 \leq 2 \cdot K - 1.$$

Proof. For a fixed u and v if $v \in S_i \setminus S_{i-1}$ then we can construct a polynomial $(x^{j_1})^{y_1} \ldots (x^{j_i})^{y_i}$ such that $v = (u^{j_1})^{y_1} \cdots (u^{j_i})^{y_i}$, where $j_k \in \{ 1, -1 \}$ and y_k's are constants from $\mathbf{G}$. This polynomial clearly has the properties of $p_{u,v}$ and the number of variable occurrences is $i \geq K_{u,v}$.

For the other direction we note that calculating these S_i sets gives us polynomials with the least variable occurrences for a function $f(x)$ with the property that $f(1) = 1$. Any 1-variable polynomial has the form $p(x) = g_1 x^{j_1} g_2 x^{j_2} g_3 x^{j_3} \ldots g_s x^{j_s} g_{s+1}$ for some s, where $j_1, \ldots, j_s \in \{ 1, -1 \}$ and $g_i \in G$. Now we alter this polynomial with the trick $g_1 x^{j_1} g_2 = g_1 x^{j_1} g_1^{-1} g_1 g_2 = (x^{j_1})^{g_1^{-1}} g_1 g_2$:

$$\begin{aligned} g_1 x^{j_1} g_2 x^{j_2} \cdots x^{j_s} g_{s+1} &= \left(x^{j_1}\right)^{g_1^{-1}} g_1 g_2 x^{j_2} \cdots x^{j_s} g_{s+1} = \\ &\left(x^{j_1}\right)^{g_1^{-1}} \left(x^{j_2}\right)^{(g_1 g_2)^{-1}} g_1 g_2 g_3 x^{j_3} \cdots x^{j_s} g_{s+1} = \cdots = \\ &\left(x^{j_1}\right)^{g_1^{-1}} \left(x^{j_2}\right)^{(g_1 g_2)^{-1}} \cdots \left(x^{j_s}\right)^{(g_1 \cdots g_s)^{-1}} g_1 \cdots g_{s+1}. \end{aligned}$$

With the notations $c_1 = g_1^{-1}, c_2 = (g_1 g_2)^{-1}, \ldots c_s = (g_1 \cdots g_s)^{-1}, c = g_1 \cdots g_{s+1}$ we have that $p(x) = (x^{j_1})^{c_1} (x^{j_2})^{c_2} \cdots (x^{j_s})^{c_s} c$. Now if $p(1) = 1$, then $c = 1$. Therefore for $s = K_{u,v}$, then there exists $c_1, \ldots, c_s$ such that $p_{u,v}$ can be realized by $p(x) = (x^{j_1})^{c_1} (x^{j_2})^{c_2} \cdots (x^{j_s})^{c_s}$, which means that $v = (u^{j_1})^{c_1} (u^{j_2})^{c_2} \cdots (u^{j_s})^{c_s}$ and $v \in S_s = S_{K_{u,v}}$.

Finally the estimation on the length is a consequence of Proposition 74. □

Remark 80. Using 1 and -1 in the exponent is slightly inconvenient, however does not make a real difference if u is conjugate to u^{-1}. We use this writing of polynomials later on.

Remark 81. Now we have an easy example that a polynomial with the least number of variable occurrences is not necessarily the shortest one for realizing a partial function: let u be a 3-cycle in $\mathbf{A}_5$, $v = u^2$, then u and v are conjugate, thus there exists $c \in \mathbf{A}_5$ such that $v = c^{-1}uc$. Hence both polynomials $c^{-1}xc$ and x^2 represent $p_{u,v}$.

Using the method described in this Section one can easily determine $K_{u,v}$'s for a given functionally complete group. In Section 3.5.2 we give quite sharp bounds on $K_{u,v}$ for certain $u, v \in \mathbf{A}_m$.

3.4.2 The partial function $f_b^{(n)}$

After investigating the function $p_{u,v}$ we move on to the more important $f_b^{(n)}$, especially to $f_b^{(2)}$.

Let $p_b^{(n)}$ be a polynomial representing $f_b^{(n)}$ such that between every two constants there is at least one variable. Using the idea of the proof of Proposition 79 the polynomial $p_b^{(n)}$ can be written as

$$p_b^{(n)}(x_1, \ldots, x_n) = \left(x_{i_1}^{j_1}\right)^{c_1} \left(x_{i_2}^{j_2}\right)^{c_2} \cdots \left(x_{i_s}^{j_s}\right)^{c_s} c_{s+1}, \tag{3.27}$$

where $i_1, \ldots, i_s \in \{1, \ldots, n\}$, $j_1, \ldots, j_s \in \{1, -1\}$, and c_r's are constants from $\mathbf{G}$. Now among $i_1, \ldots i_s$ all the elements of $\{1, \ldots, n\}$ must occur at least once, because $p_b^{(n)}$ depends on each of its variables. Now, $c_{s+1} = 1$, because $f(1, \ldots, 1) = 1$. Moreover if the ith variable occurs only once in w then if we write $x_i = b$ and $x_j = 1$ for every $j \neq i$, then we have $1 = p_b^{(n)}(1, \ldots, 1, b, 1, \ldots, 1) = b^c$ for some constant $c \in \mathbf{G}$, contradiction. Therefore we have $\left\|p_b^{(n)}\right\| \geq v\left(p_b^{(n)}\right) = s \geq 2n$ for every $n \geq 2$.

Let $A = \{r : c_r \neq 1, 1 \leq r \leq s\}$ the set of indexes of the non-identity constants. Now there is a unique partition of the set I such that every block of the partition contains only consecutive numbers and every block

is maximal in this sense. Let us denote the number of blocks with t and let us denote the blocks with A_i (where $1 \leq i \leq t$) such that if $i < j$ and $c \in A_i, d \in A_j$ arbitrary elements, then $c < d$. Let

$$s_i = |\{\, c_r : r \in A_i, r+1 \in A_i, c_r \neq c_{r+1} \,\}|\,.$$

Now it is easy to see that

$$\left\| p_b^{(n)} \right\| = s + \sum_{i=1}^{t} (2 + s_i). \tag{3.28}$$

Let $B_i = \{\, r : i_r = i \,\}$ be the index set of the variable x_i. This index set cannot contain only consecutive numbers: then $\prod_{r \in B_i} \left(x_{i_r}^{j_r}\right)^{c_r}$ would be a factor of the polynomial $p_b^{(n)}$. Since $\prod_{r \in B_i} \left(x_{i_r}^{j_r}\right)^{c_r} = f\,(1, \ldots, 1, x, 1, \ldots, 1)$ evaluates 1 for every substitution, $p_b^{(n)}$ would not depend on the variable x_i.

The number $t \geq 1$, otherwise $p_b^{(n)}$ is a term expression (containing no constants), which would imply $b = p_b^{(n)}\,(b, b, \ldots, b) = p_b^{(n)}\,(b, 1, \ldots, 1) \cdot p_b^{(n)}\,(1, b, \ldots, b) = 1$ (powers of b are interchangeable). It immediately follows that $\left\| p_b^{(n)} \right\| \geq 2n + 2$.

Let $r_1 \in B_i$ such that $c_{r_1} \neq 1$. We claim that there exist $r_2 \in B_i$, $1 \leq r_2 \leq s$, $r_2 \neq r_1$ such that $c_{r_2} \neq 1$. If there existed no such an r_2, then by $\prod_{r \in B_i} \left(x_{i_r}^{j_r}\right)^{c_r} = 1$ we can conclude to that for some k we have $x^{r_1} = x^k$ for every $x \in G$. The following lemma gives the contradiction.

Lemma 82. *Let* $\mathbf{G}$ *be a finite, simple, non-Abelian group. Then for any integer* k *and for any* $1 \neq c \in G$ *there exists* $g \in G$ *such that* $g^c \neq g^k$.

Proof. Let us suppose that for every $g \in G$ we have $g^c = g^k$. If $g \in C_{\mathbf{G}}\,(c)$, then $g^c = g$, thus $k - 1$ is divisible by the order of g. On the other hand, if $k-1$ is divisible by the order of g, then $g^k = g$, hence $g^c = g$ and $g \in C_{\mathbf{G}}\,(c)$. Therefore the subgroup $C_{\mathbf{G}}\,(c)$ is characteristic (it contains exactly those elements whose order is a divisor of $k - 1$) and hence normal. The group $\mathbf{G}$ is simple, $C_{\mathbf{G}}\,(c) \neq \mathbf{G}$, since $c \notin \{\, 1 \,\} = Z\,(\mathbf{G})$, hence $C_{\mathbf{G}}\,(c) = \{\, 1 \,\}$. This contradicts to the fact that $1 \neq c \in C_{\mathbf{G}}\,(c)$. □

If $|B_i| = 2$, e.g. $B_i = \{\, r_1, r_2 \,\}$, then $j_{r_1} = -j_{r_2}$ and $c_{r_1} = c_{r_2}$. Otherwise $\left(x_i^{j_{r_1}}\right)^{c_{r_1}} \left(x_i^{j_{r_2}}\right)^{c_{r_2}}$ can be rewritten into the form $(x_i)^c = x^k$ with some constant $c \in G$ and with an integer number k. Such equality does not hold for every $x \in G$ by Lemma 82.

Let us assume that $|B_i| = 3$ for some i, then we prove that either $t \geq 2$ or $s_j \geq 1$ for at least one $1 \leq j \leq t$. Let $B_i = \{\, r_1, r_2, r_3 \,\}$. Since

$\left(x_i^{j_{r_1}}\right)^{c_{r_1}} \left(x_i^{j_{r_2}}\right)^{c_{r_2}} \left(x_i^{j_{r_3}}\right)^{c_{r_3}} = 1$ holds for every $x \in G$, we conclude that at least one of $c_{r_1}, c_{r_2}, c_{r_3}$ is not 1. Now if only one of these three constants is different from 1, then the equation can be rewritten into the form $(x_i)^c = x^k$ with some constant $c \in G$ and with an integer number k. Such equality does not hold for every $x \in G$ by Lemma 82. If exactly two constants out of $c_{r_1}, c_{r_2}, c_{r_3}$ differs from 1, and they are the same, then we obtain a similar equation and Lemma 82 can be applied, too. If all three constant $c_{r_1}, c_{r_2}, c_{r_3}$ are equal, then the equation has a form $x_i^{j_1+j_2+j_3} = 1$, which does not hold for every $x \in G$. Therefore there are at least two constants from $c_{r_1}, c_{r_2}, c_{r_3}$ which differ from 1 and from each other, hence either $t \geq 2$ or $s_j \geq 1$.

Now if $t \geq 2$, then $\left\|p_b^{(n)}\right\| \geq 2n + 4$ by (3.28). If $t = 1$ and there exists $1 \leq i \leq n$ such that $|B_i| \geq 4$, then again $\left\|p_b^{(n)}\right\| \geq 2n + 4$. If $t = 1$ and there exists $1 \leq i \leq n$ such that $|B_i| = 3$, then $s_1 \geq 1$, hence $\left\|p_b^{(n)}\right\| \geq 2n + 4$. Therefore if $\left\|p_b^{(n)}\right\| \leq 2n + 3$, then $t = 1$, $|B_i| = 2$ for every $1 \leq i \leq n$, the constants in the form (3.27) are in one block, and either all constants are the same or there are at most two different constants. Hence we proved the following:

Proposition 83. *For every $n \geq 2$ we have*

$$2n \leq v\left(f_b^{(n)}\right) \leq V \cdot n^{\log V},$$
$$2n + 2 \leq \left\|f_b^{(n)}\right\| \leq 2 \cdot V \cdot n^{\log V} + 1,$$

where $V = v\left(f_b^{(2)}\right)$. Moreover if $\left\|f_b^{(n)}\right\| < 2n + 4$, then every variable occurs exactly twice in the shortest representation of $f_b^{(n)}$, and using the form (3.27) *there are at most two different constants.*

The lower bounds for the variable occurrences and for the length of $f_b^{(n)}$ are linear in n. On the other hand the upper bound is at least quadratic from Proposition 84. Our conjecture is that the truth is rather closer to the upper bound than the lower bound. Unfortunately there are no known methods for proving a quadratic lower bound on the length for a function over an algebra.

Now with the help of this proposition we prove that the minimal length of $f_b^{(2)}$ is at least 9. In Proposition 90 we prove that length 9 can be achieved for the group $\mathbf{A}_m$ ($m \geq 5$).

Proposition 84. *Let* $V = v\left(f_b^{(2)}\right)$. *Then we have*

$$4 \leq V \leq 4 \cdot K_{G\setminus\{1\},b},$$
$$9 \leq \left\|f_b^{(2)}\right\| \leq 8 \cdot K_{G\setminus\{1\},b} + 1.$$

Proof. Applying Proposition 83 to $n = 2$ we have that $v\left(f_b^{(2)}\right) \geq 4$ and $\left\|f_b^{(2)}\right\| \geq 6$. Let $c \in \mathbf{G}$ be a constant for which $[b, b^c] \neq 1$. Such c exists by Lemma 20. Now $p_{[b,b^c],b}([x, y^c])$ is realizing $f_b^{(2)}$, hence $v\left(f_b^{(2)}\right) \leq 4 \cdot K_{G\setminus\{1\},b}$ for some c, where $[b, b^c] \neq 1$. The upper bound for the length follows from Proposition 74. Now we only have to prove that $\left\|f_b^{(2)}\right\| \leq 8$ is not possible.

Let $p_b^{(2)}$ be a shortest representation of $f_b^{(2)}$. We deal with the different lengths separately:

Case 1: The length $\left\|f_b^{(2)}\right\| \leq 7$. By the observations which led to Proposition 83 we know that there are at most two different constants in the form (3.27). The index sets B_1 and B_2 are two-element sets, and neither of them can contain only consecutive numbers. The constants for the two occurrences of the variable x_1 have to be the same, and the constants for the two occurrences of the variable x_2 have to be the same. Moreover there must be at most one 'change' in the sequence of constants, which leaves only one possibility: $p_b^{(2)}(x_1, x_2) = \left(\left[x_1, x_2^{\pm 1}\right]^{\pm 1}\right)^c$. Now $p_b^{(2)}(b, b) = 1 \neq b$.

Case 2: The length $\left\|f_b^{(2)}\right\| = 8$. If $\left\|f_b^{(2)}\right\| = 8$ then by formula (3.28) we have the following possibilities:

1. $s = 6$, $t = 1$ and $s_1 = 0$. If $|B_i| = 3$ for any $i \in \{1, 2\}$, then $s_1 \geq 1$. Therefore either $|B_1| = 4$ and $|B_2| = 2$ or vice versa. Without loss of generality we can assume $|B_1| = 4$ and $|B_2| = 2$. $t = 1$ and $s_1 = 0$, hence there is only one constant c and it is in one block. If c conjugates any of the two occurrences of variable x_2, then it conjugates the other, too. If c conjugates both occurrences, then when calculating $p_b^{(2)}(b, b)$ we can move x_2^c and $\left(x_2^{-1}\right)^c$ next to each other. Their product is 1, therefore $p_b^{(2)}(b, b) = p_b^{(2)}(1, b) \cdot p_b^{(2)}(b, 1) = 1 \neq b$ contradiction. The same happens if c does not conjugate the occurrences of x_2, then we can move all the x_1's next to each other and have $p_b^{(2)}(b, b) = p_b^{(2)}(1, b) \cdot p_b^{(2)}(b, 1) = 1 \neq b$ contradiction.

2. $s = 5, t = 1$ and $s_1 = 1$. It is easy to see that $p_b^{(2)}(1, x) = \left(x^{j_1}\right)^{c_1} \left(x^{j_2}\right)^{c_2} x^{j_3}$ or $p_b^{(2)}(1, x) = x^{j_3} \left(x^{j_1}\right)^{c_1} \left(x^{j_2}\right)^{c_2}$ and $p_b^{(2)}(x, 1) = x^{j_4} x^{-j_4}$ (or the other

way around, let us assume it happens this way). Either way, when we calculate $f_b^{(2)}(x,x)$, using the fact that x and x^{-1} centralizes each other, we can sort the factors in such a way that the factors of $p_b^{(2)}(1,x)$ are appearing after each other, i.e.: $p_b^{(2)}(x,x) = g_1(x) \cdot p_b^{(2)}(1,x) \cdot g_2(x)$ for some terms g_1 and g_2 where $g_1(x)g_2(x) = p_b^{(2)}(x,1)$. Then $p_b^{(2)}(x,x) = g_1(x) \cdot p_b^{(2)}(1,x) \cdot g_2(x) = g_1(x) \cdot g_2(x) = p_b^{(2)}(x,1) = 1$, which contradicts with $p_b^{(2)}(b,b) = b$.

3. If $s = 4$, $t = 2$ and $s_1 = s_2 = 0$. Then $p_b^{(2)}(x,y)$ is basically $y^{j_2} \cdot (x^{j_1})^c \cdot y^{-j_2} \cdot (x^{-j_1})^c$ or $(x^{j_1})^c \cdot y^{j_2} \cdot (x^{-j_1})^c \cdot y^{-j_2}$. From $b = p_b^{(2)}(b,b)$ we can conclude to $b^{b^{\pm c}} = b^2$. Let k be the order of b. Now k is odd, as b and b^2 are conjugates, hence they have the same order. Moreover, $b^{\pm c}$ has order k, too. Now

$$b = b^1 = b^{\left(b^{\pm c}\right)^k} = b^{2^k},$$

hence $k \mid 2^k - 1$. Let p be the smallest prime divisor of k, let $k = p \cdot m$ and let t be the smallest positive integer for which $p \mid 2^t - 1$. By Fermat's Theorem we know that $2^{p-1} \equiv 1 \pmod{p}$, hence $t \mid (p-1)$. Now $2^k \equiv 1 \pmod{p}$ if and only if $t \mid k$, which means that k has a smaller prime divisor than p, as $t < p$.

4. $s = 4$, $t = 1$ and $s_1 = 2$. In this case there should be two constants c_1 and c_2 corresponding to the variables x_1 and x_2. They are ordered either as c_1, c_2, c_1, c_2 or as c_2, c_1, c_2, c_1, and we obtain $s_1 \geq 3$. The contradiction finishes the proof.

□

We can give a constant upper bound on V using the following theorem from [42]:

Theorem 85. *Let* $\mathbf{G}$ *be a finite group. Then the following are equivalent:*

1. $\mathbf{G}$ *is solvable;*
2. *no non-trivial element* g *is the product of 56 commutators of the form* $[g^h, g^k]$ *(with* $h, k \in \mathbf{G}$*);*
3. *no non-trivial 2-element* g *is the product of 126 commutators of the form* $[g^h, g^k]$ *(with* $h, k \in \mathbf{G}$*). (The element* g *is a 2-element if the order of* g *is a 2-power.)*

The following proposition is an immediate corollary of this theorem:

Proposition 86. *For every finite simple non-Abelian group* $\mathbf{G}$ *there exists* $b \in \mathbf{G}$ *such that*

$$v\left(f_b^{(2)}\right) \leq 224.$$

Moreover there exists $b \in \mathbf{G}$ *such that the order of* b *is a power of 2 and*

$$v\left(f_b^{(2)}\right) \leq 504.$$

Proof. We use the fact that if $b = \left[b^{h_1}, b^{k_1}\right] \ldots \left[b^{h_t}, b^{k_t}\right]$, then the polynomial $p(x,y) = \left[b^{h_1}, b^{k_1}\right] \ldots \left[b^{h_t}, b^{k_t}\right]$ represents the partial function $f_b^{(2)}$. The number of variable occurrences in polynomial p is $4 \cdot t$. Applying Theorem 85 finishes the proof. □

Now we can take a closer look at the results of Theorem 45 and of Theorem 75. Applying the first one to rings gives us an n factor, while Theorem 75 has a factor at least n^2 (as $V \geq 4$). The reason for that is that rings have the multiplication as a basic binary operation next to the addition, but groups have only one operation. We cannot use Theorem 48 on groups for the same reason. One wonders whether there exists another operation (corresponding to the ring multiplication) which we can take as basic operation for the group so that we obtain similar bounds as for rings or can apply Theorem 48. This is indeed the case: taking the commutator changes the algebra in a way that we can derive similar bounds to those for rings. We investigate this idea in details in Section 3.6.

3.5 The alternating group $\mathbf{A}_m$

In Section 3.1 we investigated partial functions and in Proposition 55 we stated that if a functionally complete algebra can be embedded into another one, then the length of a partial function and the number of variable occurrences for the partial function do not increase. First we prove in this Section that every finite simple non-Abelian group can be embedded into $\mathbf{A}_m$ for some m, therefore we only have to consider these groups when we are looking for shortest possible realization among finite groups. The statement holds for every finite group, so for this proposition the notation of $\mathbf{G}$ means finite group, not necessarily simple or non-Abelian.

Proposition 87. *Let* $\mathbf{G}$ *be any finite group. Then there exists* m *for which* $\mathbf{G}$ *can be embedded into* $\mathbf{A}_m$.

Proof. We can choose $m = |\mathbf{G}| + 2$, since the Cayley table of $\mathbf{G}$ gives an embedding into $S_{|\mathbf{G}|}$ and for every positive integer k there exists a subgroup

in $\mathbf{A}_{k+2}$ which is isomorphic with $\mathbf{S}_k$. This $\varphi\colon \mathbf{S}_k \to \mathbf{A}_{k+2}$ embedding is the following: for every permutation $\pi \in \mathbf{S}_k$

$$\varphi(\pi) = \begin{cases} \pi, & \text{if } \pi \text{ is even} \\ \pi \cdot (k+1, k+2), & \text{if } \pi \text{ is odd} \end{cases}.$$

Composing the two isomorphism gives us an isomorphism between $\mathbf{G}$ and a subgroup of $\mathbf{A}_m$ for $m = |\mathbf{G}| + 2$. □

In Theorem 75 we saw that one employs bounds on $v\left(f_b^{(n)}\right)$ and on the product $K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}}$ in order to obtain a proper bound on the number of variable occurrences for an arbitrary partial function f. In Proposition 93 we give a sharper quadratic bound on $v\left(f_b^{(n)}\right)$ than in Proposition 84, then we prove that b can be chosen as a 3-cycle to reach that bound. In Subsection 3.5.2 we move on to give bounds on the product $K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}}$ (Proposition 98). We summarize all the results in the following theorem:

Theorem 88. *Let $m \geq 5$ and let $N = |\mathbf{A}_m|$. Let f be an arbitrary n-ary (possibly partial) function over the group $\mathbf{A}_m$ with at most e-many non-identity values. Then the following inequalities hold:*

$$v(f) \leq \frac{1}{2} \cdot m \cdot \left(3n^2 - 3n + 2\right) \cdot \left(3N^2 - 9N + 8\right) \cdot e,$$
$$\|f\| \leq m \cdot \left(3n^2 - 3n + 2\right) \cdot \left(3N^2 - 9N + 8\right) \cdot e + 1.$$

If $4 \nmid m$, then we can replace the factor m by $\lfloor m/2 \rfloor$.

Proof. The proof follows by applying Propositions 98 and 93 below, Proposition 74 and Theorem 75. □

Let us summarize our bounds for $\mathbf{A}_m$:

Corollary 89. *Let $m \geq 5$ and let $N = |\mathbf{A}_m|$. For an arbitrary n-ary function f over $\mathbf{A}_m$ we have*

$$\|f\|_{\mathbf{A}_m} \leq m \cdot (N-1) \cdot \left(3N^2 - 9N + 8\right) \cdot \left(3n^2 - 3n + 2\right) \cdot N^{n-1} + 1.$$

If $4 \nmid m$ then we can replace the factor m by $\lfloor m/2 \rfloor$.
Moreover for every $\varepsilon > 0$ and for sufficiently large n there exists an n-ary function f_0 such that

$$\|f_0\|_{\mathbf{A}_m} \geq \frac{\log N}{1+\varepsilon} \cdot \frac{N^n}{\log n}.$$

Proof. We apply Theorem 88, Remark 76 and Theorem 46. □

3.5.1 Bounds on $v\left(f_b^{(n)}\right)$ over $\mathbf{A}_m$

From now on by $\mathbf{A}_m$ we mean the alternating group $\mathbf{A}_m$ for some $m \geq 5$. In the following propositions we determine $v\left(f_b^{(2)}\right)$ for $\mathbf{A}_m$, give some examples how b can be chosen to achieve the lowest possible $v\left(f_b^{(2)}\right)$ and give a sharp upper bound for $v\left(f_b^{(n)}\right)$.

Proposition 90. *There exists $b \in \mathbf{A}_m$ ($m \geq 5$) such that $V = v\left(f_b^{(2)}\right) = 4$ and $\left\|f_b^{(2)}\right\| = 9$.*

Proof. For $1 \neq b \in G$ there exists a polynomial $[x^{c_1}, y^{c_2}]$ (for some constants c_1 and c_2) representing $f_b^{(2)}(x, y)$ if and only if there exists a conjugate b^c such that $[b, b^c]$ is a conjugate of b. This is the case for $\mathbf{A}_m$ ($m \geq 5$) with $b = (1\,2\,3)$ or with $b = (1\,2\,3\,4\,5)$ (the multiplication is from right to left):

$$\begin{aligned}(2\,5\,3) &= [(1\,2\,3), (3\,4\,5)] \\ (1\,3\,4\,2\,5) &= [(1\,2\,3\,4\,5), (1\,5\,3\,2\,4)],\end{aligned}$$

where $(1\,5\,3\,2\,4) = (2\,4\,5)^{-1}(1\,2\,3\,4\,5)(2\,4\,5)$ and $(1\,3\,4\,2\,5) = (2\,4\,3)^{-1}(1\,2\,3\,4\,5)(2\,4\,3)$. The other conjugate relations are clear.

In the case of $\mathbf{A}_m$ for $m \geq 6$ we can even choose b from the conjugacy class of $(1\,2)(3\,4)$ or $(1\,2\,3)(4\,5\,6)$ as the following equations show:

$$\begin{aligned}(1\,4)(2\,3) &= [(1\,2)(3\,4), (2\,3)(5\,6)] \\ (1\,4\,3)(2\,5\,6) &= [(1\,2\,3)(4\,5\,6), (1\,3\,5)(2\,6\,4)].\end{aligned}$$

The conjugate relations are clear. These examples show that for every $m \geq 5$ we can choose $1 \neq b \in \mathbf{A}_m$ such that $v\left(f_b^{(2)}\right) = 4$ and $\left\|f_b^{(2)}\right\| = 9$, moreover such b can be chosen as an element of order 2 if $m \geq 6$. □

Actually, any odd cycle can be chosen as b for large enough m. For proving this we first need some preliminaries. Later on, for an element $u \in \mathbf{A}_m$ let us denote the conjugacy class of u in $\mathbf{A}_m$ with C_u and if u and v are conjugate then we use the notation introduced earlier: $u \sim u$. Let us denote the set of all permutations with the same cycle structure as u with D_u. The following lemma is quite known about conjugacy classes of $\mathbf{A}_m$ and cycle structure [4]:

Lemma 91. *Let $u_1 \in \mathbf{A}_m$. Then there exists $u_2 \in \mathbf{A}_m$ with the same cycle-structure as u_1 and u_2 is not conjugate with u_1 in $\mathbf{A}_m$ if and only if the*

cycle structure of u_1 (and u_2) contains only odd cycles with pairwise different lengths (considering 1-cycles as well). If such a u_2 exists, then for every $u_3 \in \mathbf{A}_m$ with the same cycle structure as u_1 (and u_2) we have either u_3 is conjugate to u_1 in $\mathbf{A}_m$ or u_3 is conjugate to u_2 in $\mathbf{A}_m$.

We note as an easy consequence that if $u_1 \in \mathbf{S}_m$ and $u_2 \in \mathbf{S}_m$ have the same cycle structure then u_1 is conjugate to u_2 in $\mathbf{S}_m$. For $m \geq 5$ if $u_1, u_2 \in \mathbf{A}_m$ have the same cycle structure but $u_1 \not\sim u_2$ in $\mathbf{A}_m$, then u_1 has a cycle with length at least 5. If u_1 and u_2 share the same cycle structure and u_1 stabilizes at least two points then $u_1 \sim u_2$ in $\mathbf{A}_m$.

Proposition 92. *If b is an odd cycle in $\mathbf{A}_m$ ($m \geq 5$) of length at most $\frac{2m-1}{3}$, then $v\left(f_b^{(2)}\right) = 4$ and $\left\|f_b^{(2)}\right\| = 9$.*

Proof. Let b be an arbitrary $2l+1$-cycle, where $5 \leq 3l+2 \leq m$. Without loss of generality we can suppose that $b = (1, 2, \ldots, 2l, 2l+1)$. Now let

$$u = (2l+2, 2l+3, \ldots, 3l+1, l+1, l+2, \ldots, 2l+1)\,,$$
$$v = (1, 2l+2, 2, 2l+3, 3, 2l+4, \ldots, l-1, 3l, l, 3l+1, 3l+2)\,.$$

Now $b \sim v$ as they share the same cycle structure and they stabilize at least $m-(2l+1) \geq l+1 \geq 2$ points. Hence there is a constant $c \in \mathbf{A}_m$ such that $v = b^c$. Moreover it is easy to check that $u = b^v$ and

$$b^{-1} \cdot u = (l, l-1, \ldots, 2, 1, 2l+1, 2l+2, 2l+3, \ldots, 3l+1)\,.$$

Now $b^{-1}u$ has the same cycle structure as b and stabilizes at least 2 points, hence $b \sim b^{-1} \cdot u = b^{-1} \cdot b^{b^c} = [b, b^c]$. This means $v\left(f_b^{(2)}\right) = 4$ and $\left\|f_b^{(2)}\right\| = 9$. □

We do not use this proposition later on, only that b can be chosen as a 3-cycle and for $m \geq 6$ we can choose $(1\,2)\,(3\,4)$ for b. We just mentioned this in order to show that there are many possibilities in $\mathbf{A}_m$ for choosing b in order to realize $v\left(f_b^{(2)}\right) = 4$ and $\left\|f_b^{(2)}\right\| = 9$, so we still have a chance to choose when we want to minimize the product $K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}}$ afterwards.

By Lemma 44 we already now that $v\left(f_b^{(n)}\right) \leq V \cdot n^{\log V} = 4 \cdot n^2$. Finally we give a sharper upper bound for $v\left(f_b^{(n)}\right)$ than this. The bound is still quadratic, but the constant is improved.

Proposition 93. *Let $p(x, y) = [x^{c_1}, y^{c_2}]$ with some constants from $\mathbf{G}$. Let $p^{(n)}$ be defined as in Lemma 44. Then $v\left(p^{(n)}\right) \leq 3/2 \cdot n^2 - 3/2 \cdot n + 1$.*

Proof. We prove the statement by induction on n. It is true for $n = 1, 2$: $v\left(p^{(1)}\right) = 1 \leq 3/2 \cdot (1-1)+1$, $v\left(p^{(2)}\right) = 4 \leq 3/2 \cdot (4-2)+1$ and for every $n \geq 3$

$$v\left(p^{(n)}\right) = 2 \cdot \left(v\left(p^{(\lfloor n/2 \rfloor)}\right) + v\left(p^{(\lceil n/2 \rceil)}\right)\right).$$

Let us assume that the statement is true for every $k < n$. If $n = 2l$, then

$$v\left(p^{(n)}\right) = 4 \cdot v\left(p^{(l)}\right) \leq 4 \cdot \left(3/2l^2 - 3/2l + 1\right) = \left(3/2n^2 - 3/2n + 1\right).$$

If $n = 2l+1$, then

$$\begin{aligned} v\left(p^{(n)}\right) &= 2v\left(p^{(l)}\right) + 2v\left(p^{(l+1)}\right) \\ &\leq 2 \cdot \left(3/2l^2 - 3/2l + 1\right) + 2 \cdot \left(3/2\left(l+1\right)^2 - 3/2\left(l+1\right) + 1\right) \\ &= 6l^2 + 4 \leq 6l^2 + 3l + 1 = 3/2\left(2l+1\right)^2 - 3/2\left(2l+1\right) + 1 \\ &= 3/2n^2 - 3/2n + 1. \end{aligned}$$

□

This proof shows not only that $v\left(f_b^{(n)}\right) \leq 3/2\left(n^2-n\right)+1$, but the bound is sharp for $n \leq 4$, too. For a quadratic bound we cannot expect any better as this is sharp at more than 2 points.

3.5.2 Bounds on $v\left(p_{u,v}\right)$ over $\mathbf{A}_m$

Now we know that $b \in \mathbf{A}_m$ can be chosen so that $V = v\left(f_b^{(2)}\right) = 4$. As we shown in Proposition 92 there are several choices for b. In this Section we prove that b can be chosen as a 3-cycle so that we can obtain a reasonably good (if not the best) upper bound on the product $K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}}$ in Theorem 75. First we try to bound $K_{G\setminus\{1\},(123)}$.

Lemma 94. *Let $u \in \mathbf{A}_m$ (for some $m \geq 5$) and let $D_u = C_u \cup C_{u^{-1}}$. If u is not a product of disjoint 2-cycles, then $D_u^2 = \{ u_1 \cdot u_2 \mid u_1, u_2 \in D_u \}$ contains a 3-cycle. If u is a product of disjoint 2-cycles and stabilizes at least 1 point, then D_u^2 contains a 3-cycle. If u is a product of disjoint 2-cycles and moves every m point, then D_u^2 contains a product of two disjoint 3-cycles.*

Proof. Let the longest cycle be a k-cycle in u. Without loss of generality we can assume that this cycle is the $c_k = (1, \ldots, k)$ cycle in u. If $k \leq 4$, then by Lemma 91 the conjugacy class C_u contains the elements in $\mathbf{A}_m$ with the same cycle-structure as u.

1. $k \geq 5$. Let $v = c_k^{-1}u$, $v' = v^{-1} = (1\,3)(2\,4) \cdot v^{-1} \cdot (1\,3)(2\,4)$ and let $c'_k = (2, 1, 4, 3, k, k-1, \ldots, 5) = (1\,3)(2\,4) \cdot c_k^{-1} \cdot (1\,3)(2\,4)$. Then $u' = c'_k \cdot v' \in C_{u^{-1}} \subseteq D_u$ and (multiplying from right to left)
$$u' \cdot u = c'_k v' \cdot c_k v = c'_k c_k \cdot v'v = c'_k \cdot c_k = (2\,k\,4)\,.$$

2. $k = 4$. Let $v = c_k^{-1}u$, $v' = v^{-1}$ and let $c'_k = (1\,2\,4\,3)$. Then $u' = c'_k v' \in C_u \subseteq D_u$ (since $k \leq 4$) and (multiplying from right to left) $u' \cdot u = (1\,4\,2)$.

3. $k = 3$. Let $v = c_k^{-1}u$, $v' = v^{-1}$ and let $u' = c_k \cdot v'$. Now $u' \in C_u \subseteq D_u$ (since $k \leq 4$) and (multiplying from right to left) $u' \cdot u = (1\,3\,2)$.

4. $k = 2$ and u stabilizes an element from $\{1, \ldots, m\}$. Without loss of generality we can assume that $u = (1\,2)\,v$ and stabilizes 3, then let $u' = (1\,3)\,v$. Now $u' \in C_u \subseteq D_u$ (since $k \leq 4$) and (multiplying from right to left) $u' \cdot u = (1\,2\,3)$.

5. $k = 2$ and u moves all the elements from $\{1, \ldots, m\}$. Then u is the product of 2-cycles. Without loss of generality we can assume that $u = (1\,2)\,(3\,4)\,(5\,6) \cdot v$. Let $u' = (1\,6)\,(2\,3)\,(4\,5) \cdot v$. Then $u' \in C_u \subseteq D_u$ (since $k \leq 4$) and (multiplying from right to left) $u' \cdot u = (1\,3\,5) \cdot (2\,6\,4)$.

□

The following proposition indicates what we are going to choose as b for different $\mathbf{A}_m$'s.

Proposition 95. *Let $m \geq 5$. Then*
$$K_{G\setminus\{1\},(1\,2\,3)} \leq 2, \text{ if } 4 \nmid m,$$
$$K_{G\setminus\{1\},(1\,2\,3)} \leq 4, \text{ if } 4 \mid m.$$

Proof. From Lemma 94 it is quite clear that for any $u \in \mathbf{A}_m$ we have $K_{u,(1\,2\,3)} \leq 2$ if $4 \nmid m$ and $K_{u,(1\,2\,3)} \leq 4$ if $4 \mid m$. □

Now we continue to estimate $K_{(1\,2\,3),G\setminus\{1\}}$. Let us start with a trivial observation:

Lemma 96. *Let $u = (1, \ldots, k+1)$, let $v_1 = (1, k+2, k+3, \ldots, k+l)$ and let $v_2 = (k+1, 1, k+2, k+3, k+4, \ldots, k+l-1)$. Then (multiplying from right to left)*
$$v_1 \cdot u = (1, 2, \ldots, k+l-1, k+l)$$
$$v_2 \cdot u = (1, 2, \ldots, k-1, k) \cdot (k+1, k+2, \ldots, k+l-1)\,.$$

This lemma simply shows that by multiplying with the proper l-cycle we can increase a cycle's length by $l-1$ or decrease it by 1 and create an additional cycle with length $l-1$. In the first case the resulting permutation moves $k+l$ points, in the second case it moves $k+l-1$ points. This, however, is the basic lemma on proving the following proposition.

Proposition 97. *The following inequality holds for* $\mathbf{A}_m$ *(*$m \geq 5$*):*

$$K_{(1\,2\,3),G\setminus\{1\}} \leq \lfloor m/2 \rfloor .$$

Proof. Using the idea of Lemma 96 it is easy to see (by induction) that every $2k+1$-cycle can be obtained by multiplying k-many 3-cycles. Moreover the disjoint product of an arbitrary $2k$-cycle and an arbitrary $2l$-cycle can be obtained by multiplying $k+l$-many 3-cycles. Therefore it can be proved by induction that if $u \in \mathbf{A}_m$ moves r-many points then it is a product of $\lfloor r/2 \rfloor$-many 3-cycles, which proves the inequality. □

Proposition 98. *For* $m \geq 5$ *we have*

$$K_{G\setminus\{1\},(1\,2\,3)} \cdot K_{(1\,2\,3),G\setminus\{1\}} \leq 2 \cdot \lfloor m/2 \rfloor , \text{ if } 4 \nmid m,$$
$$K_{G\setminus\{1\},(1\,2\,3)} \cdot K_{(1\,2\,3),G\setminus\{1\}} \leq 2m, \text{ if } 4 \mid m.$$

Proof. The proof is combining the results of Propositions 95 and 97. □

Finally we prove that $K_{(1\,2\,3),G\setminus\{1\}} \geq \lfloor m/2 \rfloor$:

Proposition 99. *Let* $w \in \mathbf{A}_m$ *such that* w *moves* m *points, and acts transitively on at least* $m-2$ *points (*$m \geq 5$*). If* $u_1, \ldots, u_r \in \mathbf{A}_m$ *are 3-cycles such that* $u_r \cdot u_{r-1} \cdot \cdots \cdot u_2 \cdot u_1 = w$*, then* $r \geq \lfloor m/2 \rfloor$.

Proof. First we note that $r \geq \lceil m/3 \rceil$, otherwise $u_1 \ldots u_r$ moves less than m points. Let O be the orbit with at least $m-2$ points. If m is even, then w is a product of a 2-cycle and an $m-2$-cycle. If m is odd, then w is an m-cycle.

We prove the statement by induction on m. If $m = 5$, then $r \geq \lceil 5/3 \rceil = 2$. If $m = 6$, then the only way for two 3-cycles to move all 6 points is if they are disjoint. Then they do not act transitively on at least 4 points. Hence if $m = 6$ then $r \geq 3$.

Let a_j be the number of u_i's, which contain the point j ($j = 1, \ldots, m$). Clearly $\sum_{j=1}^{m} a_j = 3r$. Let $k = |\{\, a_j \mid a_j = 1 \,\}|$. We distinguish 2 cases:

1. $k \leq r$. Now $3r = \sum_{j=1}^{m} a_j \geq 2 \cdot (m-k) + k = 2m - k \geq 2m - r$, which implies $r \geq m/2 \geq \lfloor m/2 \rfloor$.

2. $k > r$. Now there exists i_0 such that u_{i_0} moves exactly two points from $\{ a_j \mid a_j = 1 \}$ (if it contained three, then there would be a 3-orbit in w). Without loss of generality we can assume that these points are m and $m-1$. Now let $w' = u_r \cdots u_{i_0+1} u_{i_0-1} \cdots u_1$. Now $w' \in \mathbf{A}_{m-2}$, it moves $m-2$ points and acts on at least $m-4$ points transitively as taking out u_{i_0} from the product decreases the number of elements of O exactly by 2 (for elements m and $m-1$). By induction $r - 1 \geq \lfloor (m-2)/2 \rfloor = \lfloor m/2 \rfloor - 1$, hence $r \geq \lfloor m/2 \rfloor$.

□

Corollary 100. *For* $\mathbf{G} = \mathbf{A}_m$ *(*$m \geq 5$*) we have* $K_{G\setminus\{1\},b} \cdot K_{b,G\setminus\{1\}} \geq \lfloor m/2 \rfloor$.

Proof. For every u we have $K_{u,G\setminus\{1\}} \leq K_{u,b} \cdot K_{b,G\setminus\{1\}}$. Applying Proposition 99 with $u = (1\,2\,3)$ finishes the proof. □

3.6 The commutator as a basic operation

In Theorem 48 we gave an upper bound for several functionally complete algebras. Theorem 48 used some strict conditions, though, namely that there exist operations $+$ and $\cdot$ with the properties described in Theorem 6 and $\|+\| = v(+) = v(\cdot) = \|\cdot\| = 2$. This condition can be fulfilled by the Boolean algebra or rings, hence for these structures we were able to apply the theorem (Section 3.2 and Section 3.3). On the other hand, groups only have one basic binary operation: the group multiplication which corresponds to the operation $+$ mentioned above. Groups have no natural operation corresponding to the ring-multiplication $\cdot$, at least not something which has the required properties. They do have another operation, which is somehow analogous to ring multiplication: the commutator. In this Section we consider functionally complete groups when they have the commutator as an additional basic operation. We observe that the commutator indeed behaves similar to the ring multiplication. We prove Theorem 101, which gives similar bounds for the length of an arbitrary function over a two-element base set as Theorem 48 does.

Let $\mathbf{G} = (G, \cdot, ^{-1})$ be a functionally complete group and let us consider the algebra $(\mathbf{G}, [,]) = (G, \cdot, ^{-1}, [,])$ whose underlying set is G and basic operations are the group multiplication, the inverse and the commutator $[x, y] = x^{-1}y^{-1}xy$. According to the definition of length if p and q are polynomial expressions we have $\|[p, q]\| = \|p\| + \|q\|$ and $v([p, q]) = v(p) + v(q)$. The following theorem shows that using the commutator as a basic operation allows us to get rid of the $n^{\log V}$ and of the $(N-1)^{\log V}$ factors in the

upper bounds of Theorems 75 and 88 for a two-element base set. We derive a bound depending linearly on the number of non-identity values e of the function f with the same factor $n - \log e$ as in Theorem 48. This shows that the commutator seems to act similarly to the multiplication in rings or the $\wedge$ operation in the two-element Boolean algebra.

Theorem 101. *Let $\mathbf{G} = (G, \cdot, ^{-1}, 1)$ be a functionally complete group and let $\mathbf{G}^c = (\mathbf{G}, [,]) = (G, \cdot, ^{-1}, 1, [,])$, where $[,]$ is the commutator operation of $\mathbf{G}$. Let $1 \neq u \in G$, let f be an arbitrary n-ary function $f\colon \{1, u\}^n \to \{1, u\}$ with at most e-many non-identity values. Then*

$$\|f\|_{\mathbf{G}^c} \leq K_{G\setminus\{1\},u} \cdot ((10 + 3 \cdot (n - \log e)) \cdot e - 5) + 1.$$

When $\mathbf{G} = \mathbf{A}_m$ ($m \geq 5$) and u is a 3-cycle, then

$$\|f\|_{\mathbf{A}_m^c} \leq 4 \cdot ((10 + 3 \cdot (n - \log e)) \cdot e - 5) + 1.$$

If $4 \nmid m$, then we can change the constant factor 4 by 2.

In order to prove this theorem, we first have to introduce a series of non-identity elements $u_i \in G$. Let $1 \neq u \in G$ and let $u_0 = u$. We define u_n recursively: if $u_{n-1} \neq 1$ is defined, then by Lemma 20 there exists $c_{n-1} \in G$ such that $[u_{n-1}, u^{c_{n-1}}] \neq 1$. Let us fix this element c_{n-1} and let $u_n = [u_{n-1}, u^{c_{n-1}}]$. The following lemma has key importance in proving Theorem 101.

Lemma 102. *Let $\mathbf{G} = (G, \cdot, ^{-1}, 1)$ be a functionally complete group and let $\mathbf{G}^c = (\mathbf{G}, [,]) = (G, \cdot, ^{-1}, 1, [,])$, where $[,]$ is the commutator operation of $\mathbf{G}$. Let u_n be the element defined above. Let f be an arbitrary n-ary function $f\colon \{1, u\}^n \to \{1, u_n\}$ with at most e-many non-identity values ($1 \leq e \leq 2^n$). Then*

$$\|f\|_{\mathbf{G}^c} \leq (10 + 3 \cdot (n - \log e)) \cdot e - 6.$$

Proof. The idea of the proof is that using the commutator we are able to express f recursively as we did in the proof of Theorem 48. We prove the lemma by induction on n.

For $n = 1$ it is easy to see that

$$\begin{aligned}
f(x_1) &= \left[u_0, \left(c_0^{-1}u\right) x_1^{-1} c_0\right] \cdot \left[u_0, c_0^{-1} x_1 c_0\right], \text{ if } f(1) = f(u) = u_1,\\
f(x_1) &= \left[u_0, \left(c_0^{-1}u\right) x_1^{-1} c_0\right], \text{ if } f(1) = u_1,\ f(u) = 1,\\
f(x_1) &= \left[u_0, c_0^{-1} x_1 c_0\right], \text{ if } f(1) = 1,\ f(u) = u_1.
\end{aligned}$$

It is easy to see that in every case the length is at most

$$4 \cdot e \leq (10 + 3 \cdot (1 - \log e)) \cdot e - 6.$$

As for the general case, we define some new functions. Let $f_1(x_1, \ldots, x_{n-1})$ and $f_u(x_1, \ldots, x_{n-1})$ be the following $n-1$-ary functions:

$$\begin{aligned} f_1(x_1, \ldots, x_{n-1}) &= 1, \text{ if } f(x_1, \ldots, x_{n-1}, 1) = 1, \\ f_1(x_1, \ldots, x_{n-1}) &= u_{n-1}, \text{ if } f(x_1, \ldots, x_{n-1}, 1) = u_n, \\ f_u(x_1, \ldots, x_{n-1}) &= 1, \text{ if } f(x_1, \ldots, x_{n-1}, u) = 1, \\ f_u(x_1, \ldots, x_{n-1}) &= u_{n-1}, \text{ if } f(x_1, \ldots, x_{n-1}, u) = u_n. \end{aligned}$$

Now it is easy to check that

$$\begin{aligned} f(x_1, \ldots, x_n) = &\left[f_1(x_1, \ldots, x_{n-1}), \left(c_{n-1}^{-1} u\right) x_n^{-1} c_{n-1}\right] \cdot \\ &\cdot \left[f_u(x_1, \ldots, x_{n-1}), c_{n-1}^{-1} x_n c_{n-1}\right]. \end{aligned} \tag{3.29}$$

We note that if either f_1 or f_u is identically 1, then we leave out the corresponding commutator from the formula (3.29). Let f_1 have e_1-many non-identity values and let f_u have e_u-many non-identity values. If $e_1 \geq 1$ and $e_u \geq 1$ then

$$\|f\| \leq (\|f_1\| + 3) + (\|f_u\| + 3).$$

Now if both e_1 and e_u are positive then we have

$$\begin{aligned} \|f\| &\leq (e_1 \cdot (10 + 3 \cdot (n - 1 - \log e_1)) - 6 + 3) + (e_u \cdot (10 + 3 \cdot (n - 1 - \log e_u)) - 6 + 3) \\ &\leq (10 + 3 \cdot n) \cdot (e_1 + e_u) - 3 \cdot (e_1 + e_u + e_1 \cdot \log e_1 + e_u \cdot \log e_u) - 6 \\ &\leq (10 + 3 \cdot n) \cdot e - 3 \cdot e \cdot \log e - 6 \\ &\leq (10 + 3 \cdot (n - \log e)) - 6. \end{aligned}$$

Again, we use Lemma 49, just as we did in the proof of Theorem 48.

If one of e_1 and e_0 is 0, then we have

$$\begin{aligned} \|f\| &\leq e \cdot (10 + 3 \cdot (n - 1 - \log e)) - 6 + 3 \\ &\leq e \cdot (10 + 3 \cdot (n - \log e)) - 6. \end{aligned}$$

□

Proof of Theorem 101. Let f be an arbitrary function $f\colon \{1, u\}^n \to \{1, u\}$. Let f' be the n-ary function with the same domain as f and

$$\begin{aligned} f'(x_1, \ldots, x_n) &= 1, \text{ if } f(x_1, \ldots, x_n) = 1, \\ f'(x_1, \ldots, x_n) &= u_n, \text{ if } f(x_1, \ldots, x_n) = u. \end{aligned}$$

It is easy to see that $f = p_{u_n, u}(f')$. After applying Lemma 102, Proposition 79 and Corollary 42 we obtain the desired bound for a functionally complete group $\mathbf{G}$. If $\mathbf{G} = \mathbf{A}_m$, then applying Proposition 95 gives us the second bound of the theorem. □

The idea of Lemma 102 unfortunately cannot be used for an arbitrary function $f\colon G^n \to G$. We still can obtain better bounds than those in Theorem 75. The result looks similar to (3.9) in Theorem 45.

Theorem 103. *Let $\mathbf{G} = (G, \cdot, ^{-1}, 1)$ be a functionally complete group and let $\mathbf{G}^c = (\mathbf{G}, [,]) = (G, \cdot, ^{-1}, 1, [,])$, where $[,]$ is the commutator operation of $\mathbf{G}$. Let f be an arbitrary n-ary (possibly partial) function over $\mathbf{G}$ with e-many non-identity values. Let $N = |\mathbf{G}|$ and let $K = 1 + \max\left\{ K_{G\setminus\{1\},b}, K_{b,G\setminus\{1\}} \right\}$. Then the following inequalities hold:*

$$v_{\mathbf{G}^c}\left(f_b^{(n)}\right) \leq K_{G\setminus\{1\},b} \cdot n, \tag{3.30}$$

$$\left\|f_b^{(n)}\right\|_{\mathbf{G}^c} \leq K_{G\setminus\{1\},b} \cdot (3n-1) + 1 \leq 3 \cdot K_{G\setminus\{1\},b} \cdot n, \tag{3.31}$$

$$v_{\mathbf{G}^c}(\chi_{1;b}) \leq K_{G\setminus\{1\},b} \cdot v_{\mathbf{G}^c}\left(f_b^{(N-1)}\right) \leq K_{G\setminus\{1\},b}^2 \cdot (N-1), \tag{3.32}$$

$$\|\chi_{1;b}\|_{\mathbf{G}^c} \leq \left\|f_b^{(N-1)}\right\|_{\mathbf{G}^c} + v_{\mathbf{G}^c}\left(f_b^{(N-1)}\right) \cdot \max_{u\in G\setminus\{1\}} \|p_{u,b}\|_{\mathbf{G}^c}, \tag{3.33}$$

$$\|\chi_{1;b}\|_{\mathbf{G}^c} \leq 2 \cdot \left(K_{G\setminus\{1\},b} + 1\right)^2 \cdot (N-1), \tag{3.34}$$

$$\|\chi_{1;u}\|_{\mathbf{G}^c} \leq K_{b,u} \cdot \left(\|\chi_{1;b}\|_{\mathbf{G}^c} + 1\right) + 1, \tag{3.35}$$

$$\|\chi_{a_1,\ldots,a_n;b}\|_{\mathbf{G}^c} \leq \left\|f_b^{(n)}\right\|_{\mathbf{G}^c} + v_{\mathbf{G}^c}\left(f_b^{(n)}\right) \cdot \left(\|\chi_{1;b}\|_{\mathbf{G}^c} + v_{\mathbf{G}^c}(\chi_{1;b})\right), \tag{3.36}$$

$$\|\chi_{a_1,\ldots,a_n;u}\|_{\mathbf{G}^c} \leq K_{b,u} \cdot \left(\|\chi_{a_1,\ldots,a_n;b}\|_{\mathbf{G}^c} + 1\right) + 1, \tag{3.37}$$

$$\|f\|_{\mathbf{G}^c} \leq e \cdot \max\left\{ \|\chi_{a_1,\ldots,a_n;u}\|_{\mathbf{G}^c} : 1 \neq u \in G \right\}, \tag{3.38}$$

$$\|f\|_{\mathbf{G}^c} \leq 3 \cdot K^4 \cdot N \cdot n \cdot e. \tag{3.39}$$

If $\mathbf{G} = \mathbf{A}_m$ ($m \geq 5$), then

$$\|f\|_{\mathbf{A}_m^c} \leq 176 \cdot \lfloor m/2 \rfloor \cdot (N-1) \cdot n \cdot e.$$

If $4 \nmid m$, then we can replace the constant 176 by 28.

Proof. For proving (3.30) and (3.31) let us define the following sequence of group elements: $u_1 = b$ and if $u_{i-1} \neq 1$ is defined, then by Lemma 20 there exists c_{i-1} such that $[u_{i-1}, b^{c_{i-1}}] \neq 1$. Let us fix this element c_i and let $u_i = [u_{i-1}, b^{c_i}]$. We note that this sequence is the same as the sequence we defined earlier in this Section, but with a different indexing. Now let us define the following polynomials: $p^{(1)}(x_1) = x_1$ and for $i \geq 2$ let $p^{(i)}(x_1, \ldots, x_i) = [p_{i-1}(x_1, \ldots, x_{i-1}), x_n^{c_i}]$. It is easy to see that $v\left(p^{(n)}\right)_{\mathbf{G}^c} = n$ and $\left\|p^{(n)}\right\|_{\mathbf{G}^c} = 3n-2$. Now $p_{u_n,b}\left(p^{(n)}(x_1, \ldots, x_n)\right)$ realizes $f_b^{(n)}$. By Lemma 40 we have $v_{\mathbf{G}^c}\left(f_b^{(n)}\right) \leq K_{u_n,b} \cdot n$. By Corollary 42 we have $\left\|f_b^{(n)}\right\|_{\mathbf{G}^c} \leq K_{u_n,b} \cdot (3n-1) + 1$. Similarly for any function f we have $\|p_{u,v}(f)\| \leq K_{u,v} \cdot (\|f\| + 1) + 1$.

The inequalities (3.32), (3.33), (3.35), (3.36), (3.37), (3.38) follow from Lemma 40 and Corollary 42 using the following representations based on the proof of Theorem 18:

$$\begin{aligned}
\chi_{1;b}(x) &= f_b^{(N-1)}\left(bp_{u_2,b}(x)^{-1},\ldots,bp_{u_N,b}(x)^{-1}\right),\\
\chi_{1;u}(x) &= p_{b,u}\left(\chi_{1;b}(x)\right),\\
\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n) &= f_b^{(n)}\left(\chi_{1;b}\left(x_1a_1^{-1}\right),\ldots,\chi_{1;b}\left(x_na_n^{-1}\right)\right),\\
\chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n) &= p_{b,u}\left(\chi_{a_1,\ldots,a_n;b}(x_1,\ldots,x_n)\right),\\
f(x_1\ldots,x_n) &= \prod_{\substack{(a_1,\ldots,a_n)\in G^n\\ 1\neq u=f(a_1\ldots,a_n)}} \chi_{a_1,\ldots,a_n;u}(x_1,\ldots,x_n),
\end{aligned}$$

where $b\neq 1$, $u\neq 1$, $G=\{1,u_2,\ldots,u_N\}$ and $K_{u,v}=v(p_{u,v})$. The inequality 3.34 simply follows from the earlier inequalities. Then (3.39) follows from the other inequalities:

$$\begin{aligned}
&\|\chi_{a_1,\ldots,a_n;b}\|_{\mathbf{G}^c}\leq\left\|f_b^{(n)}\right\|_{\mathbf{G}^c}+v_{\mathbf{G}^c}\left(f_b^{(n)}\right)\cdot\left(\|\chi_{1;b}\|_{\mathbf{G}^c}+v_{\mathbf{G}^c}\left(\chi_{1;b}\right)\right)\\
&\leq 3\cdot(K-1)\cdot(3n-1)+(K-1)\cdot n\cdot(N-1)\cdot\left(2K^2+(K-1)^2\right)\\
&\leq 3\cdot(K-1)\cdot n\cdot\left(3+(N-1)K^2\right)-3\cdot(K-1)\\
&\leq 3\cdot K^3\cdot N\cdot n-1,\\
&\|f\|_{\mathbf{G}^c}\leq\left((K-1)\cdot\left(\|\chi_{a_1,\ldots,a_n;b}\|_{\mathbf{G}^c}+1\right)+1\right)\cdot e\leq 3\cdot K^4\cdot N\cdot n\cdot e.
\end{aligned}$$

We used in the estimations that $K\geq 2$.

If $\mathbf{G}=\mathbf{A}_m$, then we choose $b=(1\,2\,3)$. Then $K_{G\setminus\{1\},b}\leq 4$ or 2 (depending on whether $4\mid m$ or not) and $K_{b,G\setminus\{1\}}\leq\lfloor m/2\rfloor$. Therefore if $4\mid m$ then

$$\begin{aligned}
v_{\mathbf{G}^c}\left(f_b^{(n)}\right) &\leq 4n,\\
\left\|f_b^{(n)}\right\|_{\mathbf{A}_m^c} &\leq 12n-3,\\
v_{\mathbf{G}^c}\left(\chi_{1;b}\right) &\leq 16(N-1)=16N-16,\\
\|\chi_{1;b}\|_{\mathbf{A}_m^c} &\leq 16(N-1)+12(N-1)-3=28N-31,\\
\|\chi_{a_1,\ldots,a_n;b}\|_{\mathbf{A}_m^c} &\leq 12n-3+4n\cdot(44N-47)=176\cdot n\cdot(N-1)-3,\\
\|f\|_{\mathbf{A}_m^c} &\leq\left((176n(N-1)-2)\cdot m/2+1\right)\cdot e\\
&\leq 88\cdot m\cdot(N-1)\cdot n\cdot e.
\end{aligned}$$

If $4 \nmid m$ then

$$\begin{aligned}
v_{\mathbf{G}^c}\left(f_b^{(n)}\right) &\leq 2n,\\
\left\|f_b^{(n)}\right\|_{\mathbf{A}_m^c} &\leq 6n-1,\\
v_{\mathbf{G}^c}\left(\chi_{1;b}\right) &\leq 4\left(N-1\right) = 4N-4,\\
\left\|\chi_{1;b}\right\|_{\mathbf{A}_m^c} &\leq 4(N-1)+6(N-1)-1 = 10N-11,\\
\left\|\chi_{a_1,\ldots,a_n;b}\right\|_{\mathbf{A}_m^c} &\leq 6n-1+2n\cdot(14N-15) = 2\cdot n\cdot(14N-12)-1,\\
\left\|f\right\|_{\mathbf{A}_m^c} &\leq \left(2n\left(14N-12\right)\cdot\lfloor m/2\rfloor+1\right)\cdot e\\
&\leq 28\cdot\lfloor m/2\rfloor\cdot(N-1)\cdot n\cdot e.
\end{aligned}$$

□

Comparing the result of Theorem 103 to those of Theorem 88 we observe that the commutator shortens the length of the functions $f_b^{(2)}$ and $f_b^{(n)}$ to be linear in n. Therefore using the commutator improves our upper bounds on the length of an arbitrary function. Indeed, the upper bound (3.39) is now linear in n and the constant is linear in the size of the group, too. Without using the commutator our bounds in Theorem 88 are at least quadratic in these values.

We finish the Section by summarizing our bounds if the commutator is a basic operation:

Corollary 104. *Let $\mathbf{G} = (G,\cdot,^{-1},1)$ be a functionally complete group and let $\mathbf{G}^c = (\mathbf{G},[,]) = (G,\cdot,^{-1},1,[,])$, where $[,]$ is the commutator operation of $\mathbf{G}$. Let $1 \neq u \in G$, let $N = |\mathbf{G}|$ and let K be the number of conjugacy classes of $\mathbf{G}$. For every arbitrary n-ary function $f'\colon \{1,u\}^n \to \{1,u\}$ we have*

$$\left\|f'\right\|_{\mathbf{G}^c} \leq 13\cdot K\cdot(N-1)\cdot N^{n-1}.$$

When $\mathbf{G} = \mathbf{A}_m$ $(m \geq 5)$ and u is a 3-cycle, then

$$\left\|f'\right\|_{\mathbf{A}_m^c} \leq 52\cdot(N-1)\cdot N^{n-1}.$$

If $4 \nmid m$, then we can replace the constant factor 52 by 26.
For an arbitrary n-ary function f over $\mathbf{G}$ we have

$$\left\|f\right\|_{\mathbf{G}^c} \leq 3\cdot K^4\cdot(N-1)\cdot n\cdot N^n.$$

When $\mathbf{G} = \mathbf{A}_m$ $(m \geq 5)$, then

$$\left\|f'\right\|_{\mathbf{A}_m^c} \leq 176\cdot\lfloor m/2\rfloor\cdot(N-1)^2\cdot n\cdot N^{n-1}.$$

If $4 \nmid m$*, then we can replace the constant factor* 176 *by* 28.
Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *there exists an* n*-ary function* f_0 *such that*

$$\|f_0\|_{\mathbf{G}} \geq \frac{\log N}{1+\varepsilon} \cdot \frac{N^n}{\log n}.$$

Proof. We apply Theorems 101, 103, Remark 76 and Theorem 46. □

3.7 Problems

We already mentioned in Remark 38 that we do not know whether $\|f\|$ and $v(f)$ can always be realized by the same polynomial:

Problem 1. Let $\mathbf{A}$ be a functionally complete algebra. Let $f\colon A^n \to A$ be an arbitrary function with domain A^n. Does a polynomial p exist over the algebra $\mathbf{A}$ such that $v(p) = v(f)$ and $\|p\| = \|f\|$?

In Section 3.4.2 we observed that there is a gap between the linear lower bound and the at least quadratic upper bound for the functions $f_b^{(n)}$. We conjecture that a quadratic lower bound can be found, but there are no methods for proving such a lower bound.

Problem 2. With what rate do $v\left(f_b^{(n)}\right)$ and $\left\|f_b^{(n)}\right\|$ increase in n?

Chapter 4

Computations over functionally complete groups

In Chapter 3 we investigated the length of polynomials over functionally complete groups. We gave several upper bounds on the length of realizing polynomials for an arbitrary n-ary function. A natural question is to ask how efficient these realizations are. From the practical perspective, though, length of the polynomials is not necessarily the best measure for efficiency.

Nowadays, in the age of computers, the most frequent problems are the time and resource needs of different calculations. In this Chapter by 'efficiency' we mean required computational time. To be precise, we need to fix a computational model. We consider two models in this Chapter: acyclic circuits over an algebra and finite-state sequential circuits over simple non-Abelian groups.

In our first approach we investigate the *complexity of circuits*. For a functionally complete algebra $\mathbf{A}$, an $\mathbf{A}$-circuit is essentially a directed acyclic digraph with labelled nodes. The source nodes are labelled by variables or by constants, the other nodes (called 'gates') are labelled by basic operations of $\mathbf{A}$. A calculation at a gate is the application of the corresponding basic function on the values calculated at the sources of the incoming edges. Therefore a circuit computes a function at every gate. If every calculation at a gate takes one time-step, then the number of gates corresponds to the required time for calculating a function with a single processor machine. Similarly a longest path corresponds to the required time calculating a function with a multiple processor machine.

In the Section 4.2 we find circuits computing an arbitrary function over a functionally complete group using the ideas of Section 2.3. Then we compare the functionally complete groups (especially the alternating groups) to other functionally complete algebras in the terms of circuit complexity. We

investigate the case where the other algebra is either one of the two-element algebras $\mathbf{B}$ and $\mathbf{B}_0$ (in Section 4.3) or a ring (in Section 4.4). In particular we investigate the case when the ring is a field of prime order (in Section 4.4).

Later in Section 4.5 we consider a completely different model: the notion of *finite-state sequential circuits* over simple non-Abelian groups, and investigate its efficiency.

4.1 Circuit complexity

The notion of circuit complexity emerged from the idea of finding functions which can be calculated faster than others. Our main reference on circuit complexity are the books [33] and [40].

Definition 105. Let $\mathbf{A} = (A, g_1, \ldots, g_m)$ be an algebra with underlying set A and basic operations $g_1, \ldots, g_m$. An n-ary $\mathbf{A}$-*circuit* C consists of inputs $x_1, \ldots, x_n$ and finitely many gates $G_1, \ldots, G_b$. The gate G_i is a $(n_i + 1)$-tuple $(h_i, P_1, \ldots, P_{n_i})$ such that h_i is an n_i-ary basic operation of $\mathbf{A}$ and $P_1, \ldots, P_{n_i}$ are predecessors from the set $A \cup \{ x_1, \ldots, x_n \} \cup \{ G_1, \ldots, G_{i-1} \}$. We denote by Res_{G_i} the function computed at the gate G_i. We define Res inductively on an arbitrary input $\bar{x} = (x_1, \ldots, x_n) \in A^n$. For an input variable x_i let $\mathrm{Res}_{x_i}(\bar{x}) = x_i$, for a constant $a \in A$ let $\mathrm{Res}_a(\bar{x}) = a$. For $G_i = (h_i, P_1, \ldots, P_{n_i})$ let $\mathrm{Res}_{G_i}(\bar{x}) = h_i\left(\mathrm{Res}_{P_1}(\bar{x}), \ldots, \mathrm{Res}_{P_{n_i}}(\bar{x})\right)$. Finally the output of the circuit is a vector $(y_1, \ldots, y_k)$, where every y_i is an input variable, or a constant, or a gate. This represents the function $f\colon A^n \to A^k$ computed by the circuit, i.e. $f = (f_1, \ldots, f_k)$ such that f_i is the function $\mathrm{Res}_{y_i}(\bar{x})$ computed at y_i.

Remark 106. As we already mentioned in Section 3.1, sometimes one has to work with partial functions instead of fully defined ones. The notion of a circuit computing a function can be naturally extended to partial functions: let us assume that C computes a function $f\colon A^n \to A^k$. Let $g\colon A^n \to A^k$ be a partial function with domain set D. Let us assume that $f\big|_D = g\big|_D$. Then we say that the circuit C computes the partial function g. Moreover, it is clear that if an algebra $\mathbf{A}$ is functionally complete, then for every (possibly partial) function f can be computed by an $\mathbf{A}$-circuit.

Remark 107. A circuit differs in an essential way from the rooted tree corresponding to a polynomial. In a circuit, intermediate results of gates can be used by multiple other gates further 'downstream', rather than only once. Thus circuits may be viewed as a generalization of polynomials.

It is easy to represent a circuit as a directed acyclic graph with nodes labelled by the basic operations of $\mathbf{A}$, variables, and constants. The source

nodes correspond to inputs and to constants, the other nodes correspond to the gates. Let us label the node corresponding to variable x_i by x_i. Let us label the node corresponding to variable c by c. Let us label the node corresponding to G_i by the basic function h_i. There is an edge going from every predecessor of G_i to the node corresponding to G_i. The incoming edges at the node G_i are ordered, where this ordering represents the ordering of the inputs of h_i.

This circuit model is quite close to how computers calculate different functions. If we assume that each gate-computation takes one time-step, then computing f for a particular input using a circuit C takes $s(C)$-many time-steps with a single processor. If, however, one can do arbitrary many computations parallelly (by having multiple processors) then computing f for a particular input using C takes $d(C)$-many time-steps. Therefore the size corresponds to the required time for single processor computations, while the depth corresponds to the required time for multi-processor computations.

We want to compare the efficiency of circuits which calculate particular functions over different functionally complete algebras. First we need some way to measure this efficiency.

Definition 108. The *size* or *complexity* $s(C)$ of a circuit C is the number of gates in C. The *depth* $d(C)$ of the circuit C is the length of the longest path in C. For a function $f\colon A^n \to A^k$ let the *complexity of f with respect to* $\mathbf{A}$ be the size of a smallest n-ary $\mathbf{A}$-circuit which computes f; let the *depth of f with respect to* $\mathbf{A}$ be the depth of an n-ary $\mathbf{A}$-circuit which computes f and has the smallest depth:

$$s_{\mathbf{A}}(f) = \min\{\, s(C) : C \text{ computes } f \text{ over } \mathbf{A} \,\},$$
$$d_{\mathbf{A}}(f) = \min\{\, d(C) : C \text{ computes } f \text{ over } \mathbf{A} \,\}.$$

When it does not create confusion, we omit the subscript and just write $s(f)$ for the size and $d(f)$ for the depth.

Remark 109. We defined circuits representing an $f\colon A^n \to A^k$ function. Throughout the thesis we only consider $A^n \to A$ functions, unless explicitly indicated otherwise. This is not an essential restriction, as for a function $f\colon A^n \to A^k$ we have $f = (f_1, \ldots, f_k)$, where $f_i\colon A^n \to A$. Now it is easy to see that

$$\max_{1\le i\le k} s(f_i) \le s(f) \le \sum_{i=1}^{k} s(f_i),$$
$$d(f) = \max d(f_i).$$

Indeed, a circuit C computing f in particular computes every f_i $(1 \leq i \leq k)$. On the other hand if circuits $C_1, \ldots C_k$ compute the functions $f_1, \ldots, f_k$, then their union computes f.

Remark 110. It is easy to see that constant functions or projections can be represented by a circuit without any gates, therefore their size and depth is 0.

We now introduce a definition for technical purposes. We do not want to change the usual complexity measure. We use the notions of non-unary size and non-unary depth for giving upper and lower bounds on the size and on the depth.

In Chapter 3 we mentioned that the length of a polynomial is the same as the number of leaves of the corresponding branching tree. This branching tree can be considered as a circuit. There are some differences, though. The main difference is that in the branching tree every node represents an at least binary basic function. The edges are labelled with compositions of unary functions. We can easily obtain a circuit from a branching tree by replacing every edge with its correspondent chain of unary gates. With this method we can assign a circuit to every branching tree. Let us call this circuit the *circuit corresponding to the branching tree*.

We have to observe, though, that due to the unary basic operations, the relationship is not clear either between the sizes or between the depths of the branching tree and of the corresponding circuit. This idea suggests the elimination of the unary part of a circuit, just like how we obtained a branching tree from a rooted tree in Chapter 3. We collapse every chain of unary basic operations into a single edge, and we consider the size and the depth of the obtained circuit. The precise definition is the following.

Definition 111. Let C be an $\mathbf{A}$-circuit. Let C^* be the circuit which we obtain from C by removing every unary gate: if G_i is a unary gate with predecessor P, then we remove the gate G_i, and whenever G_i was a predecessor of any other gate, then we change that predecessor to P. By iterating this method we obtain a circuit C^*, which has no unary gates. This circuit does not necessarily compute the same function as C, but they are related.

The *non-unary size* or *non-unary complexity* $s^*(C)$ of a circuit C is the number of gates in C^*. The *non-unary depth* $d^*(C)$ of the circuit C is the length of the longest path in C^*. For a function $f\colon A^n \to A^k$ let the *non-unary complexity of f with respect to* $\mathbf{A}$ be the non-unary size of a smallest n-ary $\mathbf{A}$-circuit which computes f; let the *non-unary depth of f with respect to* $\mathbf{A}$ be the non-unary depth of an n-ary $\mathbf{A}$-circuit which computes f and

has the smallest depth:

$$s^*_{\mathbf{A}}(f) = \min\{\, s^*(C) : C \text{ computes } f \text{ over } \mathbf{A} \,\},$$
$$d^*_{\mathbf{A}}(f) = \min\{\, d^*(C) : C \text{ computes } f \text{ over } \mathbf{A} \,\}.$$

When it does not create confusion, we omit the subscript and just write $s^*(f)$ for the non-unary size and $d^*(f)$ for the non-unary depth.

It is clear that the depth of a branching tree is essentially the same as the non-unary depth of the corresponding circuit. We reveal more about the relationship of these quantities. For that we need to introduce some more notations.

Let $\mathbf{A} = (A, g_1, \ldots, g_m)$ be a functionally complete algebra with underlying set A and basic operations $g_1, \ldots, g_m$. Let $g_0 = id$ the identity function over A. Let us suppose that the functions $g_0, \ldots, g_{m_0}$ are unary, the functions $g_{m_0+1}, \ldots, g_m$ are at least binary. Then let us denote the unary part of the algebra by $\mathbf{A}^1$, i.e. $\mathbf{A}^1 = (A, g_0, \ldots, g_{m_0})$. Let H be the unary functions which can be represented as polynomials over $\mathbf{A}^1$ (including the identity function $id : x \mapsto x$). Let

$$U = \max_{f \in H} s_{\mathbf{A}^1}(f).$$

Note that if $H = \{\, id \,\}$, then $U = 0$.

Proposition 112. *Let $\mathbf{A}$ be a functionally complete algebra, where every basic operation is at most k-ary ($k \geq 2$). Let U be the number defined above. Then for any arbitrary n-ary (possibly partial) function f over $\mathbf{A}$ we have*

$$s^*(f) \leq s(f) \leq s^*(f) + (k+1) \cdot U \cdot s^*(f),$$
$$d^*(f) \leq d(f) \leq d^*(f) + U \cdot (d^*(f) + 1).$$

Proof. It is clear that $s^*(f) \leq s(f)$ and $d^*(f) \leq d(f)$. Let us assume that C_1 is an $\mathbf{A}$-circuit which computes f and $s(C_1) = s(f)$. Let C_1^* be the circuit we obtain from C_1 by collapsing every chain of unary basic operations as in Definition 111. If a chain contains more than U-many unary functions, then this chain can be replaced by a chain of at most U-many basic unary functions (by the definition of U). This way the size of C_1 can be decreased. Therefore every chain contains at most U-many unary basic functions.

In C_1^* there are $s^*(f)$-many gates labelled by an at least binary basic operation. Each of the gates has at most k-many incoming edges, which represent (possibly empty) chains of basic unary functions. Moreover every gate of C_1^* might have been a predecessor of a unary chain. As every chain contains at most U-many basic unary operations, we can conclude that we removed from C at most $(k+1) \cdot U \cdot s^*(f)$-many edges.

We can derive the upper bound for $d(f)$ similarly: let C_2 be an $\mathbf{A}$-circuit which computes f and $d(f) = d(C_2)$. Without loss of generality we can assume that every unary chain in C_2 contains at most U-many basic unary functions, otherwise we change the particular chain with an at most U-long chain. Now collapse every unary chain and obtain the circuit C_2^* as in Definition 111. The longest path in C_2 can contain at most $d^*(f)$-many gates labelled with a non-unary function. Each of the gates have incoming edges, which represent (possibly empty) chains of basic unary functions. Moreover every gate of C_2^* might have been a predecessor of a unary chain. As every chain contains at most U-many basic unary operations, we can conclude that in the longest path there are at most $(d^*(f)+1)\cdot U$-many unary gates, which proves the last inequality. □

This proposition shows that it is important how the basic operations of a functionally complete algebra are defined. Therefore we set that the basic operations of a ring are the binary operations $+$, $-$ and $\cdot$. The basic operations of a group are the binary multiplication and the unary inverse. The basic operations of the two-element Boolean algebra $\mathbf{B}$ are the unary negation, and the binary $\wedge$ and $\vee$. The basic operations of the two-element algebra $\mathbf{B}_0$ are the binary NAND and NOR.

Corollary 113. *Let $\mathbf{A}$ be a functionally complete algebra and let f be an arbitrary function over $\mathbf{A}$. If $\mathbf{A}$ is a functionally complete ring or $\mathbf{A}$ is the two-element algebra $\mathbf{B}_0$, then $s(f) = s^*(f)$ and $d(f) = d^*(f)$. If $\mathbf{A}$ is a functionally complete group or the two-element Boolean algebra $\mathbf{B}$, then*

$$s^*(f) \leq s(f) \leq 4 \cdot s^*(f),$$
$$d^*(f) \leq d(f) \leq 2 \cdot d^*(f) + 1.$$

Proof. Functionally complete rings and $\mathbf{B}_0$ has no unary operations. The two-element Boolean algebra $\mathbf{B}$ and the groups have one unary operation which has order two, therefore $U = 1$. Every other basic operation is binary, hence $k = 2$. Applying Proposition 112 finishes the proof. □

In the following we give some bounds on the size, on the depth, on the non-unary size, and on the non-unary depth of an arbitrary function. Generally it is easier to obtain lower bounds on the size or on the depth, and it is easier to obtain upper bounds on the non-unary size or on the non-unary depth. First we give bounds on the non-unary size and on the non-unary depth by having information on the length.

Proposition 114. *Let* $\mathbf{A}$ *be a functionally complete algebra, where every basic operation is at most* k*-ary* ($k \geq 2$)*. Then for any arbitrary* n*-ary (possibly partial) function* f *over* $\mathbf{A}$ *we have*

$$\lceil \log_k \|f\| \rceil \leq d^*(f) \leq s^*(f) \leq \|f\| - 1.$$

Proof. The inequality $d^*(f) \leq s^*(f)$ is trivial. Let p be a polynomial realizing f over $\mathbf{A}$ such that $\|f\| = \|p\|$. This polynomial can be represented by a rooted tree. Let us consider an $\mathbf{A}$-circuit corresponding to the rooted tree of p. This circuit contains at most $\|p\| - 1$-many non-unary gates, since p contains at most $\|p\| - 1$-many occurrences of non-unary basic operations. Therefore $s^*(f) \leq \|f\| - 1$.

All that remains is to prove that $\lceil \log_k \|f\| \rceil \leq d^*(f)$ holds. Let C be an $\mathbf{A}$-circuit which computes f with non-unary depth $d^*(C) = d^*(f)$. Then the circuit can be translated to a rooted tree with the same depth, which rooted tree corresponds to a polynomial p'. The longest path in the rooted tree has $d^*(f)$-many branching nodes, therefore the tree has at most $k^{d^*(f)}$-many leaves. This proves that $\|f\| \leq \|p'\| \leq k^{d^*(f)}$, hence $\log_k \|f\| \leq d^*(f)$. Since $d^*(f)$ is an integer number, we have $\lceil \log_k \|f\| \rceil \leq d^*(f)$. □

Proposition 115. *For functions* $f, g_1, \ldots, g_n$ *we have that*

$$s(f(g_1, \ldots, g_n)) \leq s(f) + \sum_{i=1}^{n} s(g_i),$$
$$d(f(g_1, \ldots, g_n)) \leq d(f) + \max_{1 \leq i \leq n} d(g_i),$$

Proof. Let $C, C_1, \ldots, C_n$ be circuits computing $f, g_1, \ldots, g_n$ respectively, such that $s(C) = s(f)$ and $s(C_i) = s(g_i)$ for every $1 \leq i \leq n$. Now by replacing in C every variable x_i by the circuit C_i we obtain a circuit of size $s(f) + \sum_{i=1}^{n} s(g_i)$ which computes the function $f(g_1, \ldots, g_n)$.

For the inequality about the depth, let $C', C'_1, \ldots, C'_n$ be circuits computing $f, g_1, \ldots, g_n$ respectively, such that $d(C') = d(f)$ and $d(C'_i) = d(g_i)$ for every $1 \leq i \leq n$. Now by replacing in C' every variable x_i by the circuit C'_i we obtain a circuit of depth $d(f) + \max_{1 \leq i \leq n} d(g_i)$ which computes the function $f(g_1, \ldots, g_n)$. □

The following lemma plays a similar role as Lemma 44, and determines the sufficient size and depth for iterating a binary function.

Lemma 116. *Let* f *be a binary function over an algebra* $\mathbf{A}$*. Let us define the following series of functions:* $f^{(1)}(x_1) = x_1$, $f^{(2)}(x_1, x_2) = f(x_1, x_2)$ *and for every integer* $n \geq 2$*:*

$$\begin{aligned} f^{(2n-1)}(x_1,\ldots,x_{2n-1}) &= f\left(f^{(n)}(x_1,\ldots,x_n), f^{(n-1)}(x_{n+1},\ldots,x_{2n-1})\right) \\ f^{(2n)}(x_1,\ldots,x_{2n}) &= f\left(f^{(n)}(x_1,\ldots,x_n), f^{(n)}(x_{n+1},\ldots,x_{2n})\right). \end{aligned}$$

Let $S = s(f)$ *and* $D = d(f)$. *Then*

$$\begin{aligned} s\left(f^{(n)}\right) &\leq (n-1)\cdot S, \\ d\left(f^{(n)}\right) &\leq \lceil \log n \rceil \cdot D. \end{aligned}$$

Proof. We prove the lemma by induction on n. Both inequalities trivially hold for $n = 1, 2$. Let us suppose that the inequalities hold for every $k < n$. Now using the recursive definition of $f^{(n)}$ and Proposition 115 we have

$$\begin{aligned} s\left(f^{(n)}\right) &\leq s\left(f^{(2)}\right) + s\left(f^{\lfloor n/2 \rfloor}\right) + s\left(f^{\lceil n/2 \rceil}\right) \\ &\leq (2 - 1 + \lfloor n/2 \rfloor - 1 + \lceil n/2 \rceil - 1)\cdot S \\ &\leq (n-1)\cdot S. \end{aligned}$$

Similarly we have

$$\begin{aligned} d\left(f^{(n)}\right) &\leq d\left(f^{(2)}\right) + \max\left\{ d\left(f^{\lfloor n/2 \rfloor}\right), d\left(f^{\lceil n/2 \rceil}\right)\right\} \\ &\leq (1 + \lceil \log \lceil n/2 \rceil \rceil)\cdot D \\ &\leq \lceil \log n \rceil \cdot D. \end{aligned}$$

□

Now we are ready to give bounds on the size and on the depth of an arbitrary function.

Theorem 117. *Let* $\mathbf{A}$ *be a functionally complete algebra,* $N = |A|$. *Let* $0, 1 \in A$ *be two distinct elements and let* $+, \cdot, \chi_a$ *be functions with properties such as in Theorem 6. Let* $\chi_{a_1,\ldots,a_n}$ *be the characteristic function for the* n*-tuple* $(a_1,\ldots,a_n)$. *Let us suppose that* S, D *are positive real numbers such that* $S \geq \max\{ s(\chi_a) : a \in A \}$ *and* $D \geq \max\{ d(\chi_a) : a \in A \}$. *Let* f *be an arbitrary* n*-ary function over* $\mathbf{A}$ *with* e*-many non-zero values, where* $1 \leq e \leq |A|^n$. *Then the following inequalities hold:*

$$s(\chi_{a_1,\ldots,a_n}) \leq (n-1)\cdot s(\cdot) + \sum_{i=1}^{n} s(\chi_{a_i}) \leq n\cdot(S + s(\cdot)) - s(\cdot), \tag{4.1}$$

$$d(\chi_{a_1,\ldots,a_n}) \leq \lceil \log n \rceil \cdot d(\cdot) + \max_{1\leq i\leq n} d(\chi_{a_i}) \leq \lceil \log n \rceil \cdot d(\cdot) + D, \tag{4.2}$$

$$s(f) \leq (e-1)\cdot s(+) + e\cdot\left(s(\cdot) + \max_{a_i \in A} s(\chi_{a_1,\ldots,a_n})\right), \tag{4.3}$$

$$d(f) \leq \lceil \log e \rceil \cdot d(+) + d(\cdot) + \max_{(a_1,\ldots,a_n)\in A^n} d(\chi_{a_1,\ldots,a_n}). \tag{4.4}$$

If $N \geq 3$, then

$$s(f) \leq ((3 + n - \log_N e) \cdot e - 2) \cdot (s(+) + s(\cdot) + S), \tag{4.5}$$
$$s(f) \leq e \cdot (s(+) + n \cdot s(\cdot) + n \cdot S) - s(+), \tag{4.6}$$
$$d(f) \leq \lceil \log e \rceil \cdot d(+) + (1 + \lceil \log n \rceil) \cdot d(\cdot) + D. \tag{4.7}$$

If $N = 2$, then

$$s(f) \leq ((3 + n - \log_N e) \cdot e - 2) \cdot (s(+) + s(\cdot) + S), \tag{4.8}$$
$$s(f) \leq e \cdot (s(+) + (n-1) \cdot s(\cdot) + n \cdot S) - s(+), \tag{4.9}$$
$$d(f) \leq \lceil \log e \rceil \cdot d(+) + \lceil \log n \rceil \cdot d(\cdot) + D. \tag{4.10}$$

Proof. The inequalities apart from (4.5) and (4.8) follow from simply applying Proposition 115 and Lemma 116 on the following representations:

$$\chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n) = \prod_{i=1}^{n} \chi_{a_i}(x_i),$$
$$f(x_1,\ldots,x_n) = \sum_{(a_1,\ldots,a_n)\in A^n} (f(a_1,\ldots,a_n) \cdot \chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n)),$$

and whenever the algebra has only 2 elements, then

$$f(x_1,\ldots,x_n) = \sum_{(a_1,\ldots,a_n)\in A^n} \chi_{a_1,\ldots,a_n}(x_1,\ldots,x_n).$$

Here we consider $\prod$ and $\sum$ as the iterated versions of $\cdot$ and $+$ in the way we described in Lemma 116.

The inequalities (4.5) and (4.8) are the same. The proof of the inequality (4.5) is rather similar to the one for Theorem 48 in Chapter 3. We prove the inequality (4.5) by induction on n. If $n = 1$, then $f(x) = \sum_{a\in A} f(a) \cdot \chi_a(x)$, which has size at most $e \cdot (s(\cdot) + S) + (e-1) \cdot s(+) \leq ((3 + 1 - \log_N e) \cdot e - 2) \cdot (s(+) + s(\cdot) + S)$ if we do not use any of those summands where $f(a) = 0$.

The idea of the proof is that we try to calculate f recursively. For every element $a \in A$ let f_a be an $n-1$-ary function, such that $f_a(x_1,\ldots,x_{n-1}) = f(x_1,\ldots,x_{n-1},a)$. Now $f(x_1,\ldots,x_n) = \sum_{a\in A} f_a(x_1,\ldots,x_{n-1}) \cdot \chi_a(x_n)$. Let f_a have e_a-many non-zero values. Let $T = s(\cdot) + s(+) + S$. Now we apply the induction hypothesis for the $n-1$-ary functions. If there is only one $e_a > 0$, then $e_a = e$ and

$$\begin{aligned} s(f) &\leq s(f_a) + s(\cdot) + S + s(+) = s(f_a) + T \\ &\leq e \cdot T \cdot (3 + n - 1 - \log_N e) - 2 \cdot T + T \\ &\leq e \cdot T \cdot (3 + n - \log_N e) - 2 \cdot T. \end{aligned}$$

Otherwise

$$\begin{aligned}\|f\| &\leq \sum_{a\in A}\left(s\left(f_a\right)+s\left(\cdot\right)+S+s\left(+\right)\right)=\sum_{a\in A}\left(s\left(f_a\right)+T\right)\\ &\leq \sum_{e_a>0}\left(e_a\cdot T\cdot\left(3+n-1-\log_N e_a\right)-2\cdot T+T\right)\\ &\leq \sum_{e_a>0} e_a\cdot T\cdot(3+n)-T\cdot\left(\sum_{e_a>0} e_a+\sum_{e_a>0} e_a\log_N e_a\right)-\sum_{e_a>0} T\\ &\leq e\cdot T\cdot(3+n)-T\cdot e\cdot\log_N e-2\cdot T\\ &= e\cdot T\cdot\left(3+n-\log_N e\right)-2\cdot T.\end{aligned}$$

The last inequality holds by Lemma 49. □

Remark 118. While the idea of Theorem 48, namely iterate functions recursively, can be used for giving sharper bounds on the size, it cannot be used for building efficient circuits minimizing the depth. We note that if e is large, e.g. $e \geq c_1 \cdot N^{n-c_2}$, then bounds (4.5) and (4.8) are linear in e and S, while bounds (4.6) and (4.9) are linear not only in e and S, but in n, too. On the other hand if e is small, e.g. $e \leq c_3 \cdot N^{c_4 \cdot n}$ (for some $c_4 < 1$), then all bounds (4.5), (4.6), (4.8) and (4.9) are linear in e, n and S.

Unfortunately Theorem 117 cannot be applied to functionally complete groups. It can be applied to functionally complete rings, or to the two-element algebras $\mathbf{B}$ and $\mathbf{B}_0$ as the following Corollary shows. We prove some upper bounds on the size and on the depth of an arbitrary function over a functionally complete group in Section 4.2.

Corollary 119. *Let* $\mathbf{A}$ *be a functionally complete ring or any of the two-element algebras* $\mathbf{B}_0$ *or* $\mathbf{B}$. *Let* $N = |A|$. *Let us suppose that* S, D *are positive real numbers such that* $S \geq \max\{\, s\left(\chi_a\right) : a \in A \,\}$ *and* $D \geq \max\{\, d\left(\chi_a\right) : a \in A \,\}$. *Let* f *be an arbitrary* n*-ary function over* $\mathbf{A}$ *with* e*-many non-zero values, where* $1 \leq e \leq |A|^n$. *Then*

$$\begin{aligned}s\left(f\right) &\leq \left(\left(3+n-\log_N e\right)\cdot e-2\right)\cdot\left(S+2\right),\\ d\left(f\right) &\leq \lceil\log e\rceil+\lceil\log n\rceil+D+1.\end{aligned}$$

Moreover, if $N = 2$ *then*

$$\begin{aligned}s\left(f\right) &\leq 3\cdot e\cdot\left(3+n-\log e\right)-6,\\ d\left(f\right) &\leq \lceil\log e\rceil+\lceil\log n\rceil+1.\end{aligned}$$

Proof. The first two inequalities are simple consequence of Theorem 117.

If $N = 2$, then $\mathbf{A}$ is one of the three algebras $\mathbf{B}$, $\mathbf{B}_0$, and $\mathbf{Z}_2$. In any case we have $S = D = 1$. The inequalities for the case of $N = 2$ are now an easy consequence of Theorem 117. □

The following theorem gives a lower bound on the size and on the depth:

Theorem 120. *Let $\mathbf{A}$ be a functionally complete algebra. Let us suppose that every basic operation is at most k-ary. For every $\varepsilon > 0$ and for sufficiently large n (depending on ε) there exists an n-ary function f_1 over $\mathbf{A}$ such that*

$$s(f_1) \geq \frac{1}{k-1+\varepsilon} \cdot \frac{|A|^n}{n}.$$

Moreover for every $\varepsilon > 0$ and for sufficiently large n (depending on ε) there exists an n-ary function f_2 over $\mathbf{A}$ such that

$$d(f_2) \geq \frac{\log|A|}{\log k} \cdot n - \frac{1}{\log k} \cdot \log\log n + \frac{\log\log|A| - \log(1+\varepsilon)}{\log k}.$$

Proof. The lower bound for the depth follows immediately from Propositions 112, 114 and Theorem 46.

As for the size we use a similar counting idea as Theorem 46 in Chapter 3. Let us consider the number of at most n-ary circuits which have size at most s. Let this number be $N(s)$. If S is the least number such that all n-ary functions have size at most S, then $N(S) \geq |A|^{|A|^n}$. This gives us a lower bound on the size.

Let $\mathbf{A}$ have m-many basic operation symbols. Let us consider an arbitrary $\mathbf{A}$-circuit with size s. Every gate can be labelled by m-many basic operations, hence for every circuit the labelling of the gates can be done at most m^s-many ways. There are at most $n+|A|+s-1$-many possibilities to choose one predecessor of a gate (namely the predecessor is one of the variables, or one of the constants, or one of the other $s-1$ gates). There are at most k-many predecessors for every gate, hence there are at most $(s+n+|A|-1)^{ks}$-many ways to choose every predecessor for every gate. If a circuit has s-many gates, then it computes at most s-many functions at its gates. Moreover every circuit with size s has been counted $s!$-many times, namely for the different numberings for the gates. Therefore we have

$$N(s) \leq (s+n+|A|-1)^{ks} \cdot m^s \cdot s \cdot (s!)^{-1}.$$

Let f be an n-ary function such that it has the largest size. Let $S = s(f)$. Now applying $N(S) \geq |A|^{|A|^n}$ we have that

$$|A|^n \cdot \log|A| \leq k \cdot S \cdot \log(S+n+|A|-1) + S \cdot \log m + \log S - \log S!.$$

By Stirling formula (see e.g. [32]), $S! \geq c_0 \cdot S^{S+1/2} \cdot e^{-S}$, where $c_0 = \sqrt{2\pi}$ and e is the natural base. Now

$$|A|^n \cdot \log |A| \leq k \cdot S \cdot \log (S + n + |A| - 1) + S \cdot \log m + \log S + S \cdot \log e \\ - (S + 1/2) \cdot \log S - \log c_0.$$

Since the lefthand-side of the inequality is exponential in n, and the righthand-side is polynomial in n and in S, for sufficiently large n we have $n+|A|-1 \leq S$. Now we have

$$|A|^n \cdot \log |A| \leq (k-1) \cdot S \cdot \log S + (k + \log m + \log e) \cdot S + 1/2 \cdot \log S.$$

For sufficiently large n we have $k+\log m+\log e \leq \varepsilon/2 \cdot \log S$ and $1/2 \leq \varepsilon/2 \cdot S$. Thus we obtain

$$\frac{\log |A|}{k - 1 + \varepsilon} \cdot |A|^n \leq S \cdot \log S.$$

Let $c = \frac{1}{k-1+\varepsilon}$. Now if $S < c \cdot |A|^n / n$, then for sufficiently large n we have

$$\begin{aligned} S \cdot \log S &< c \cdot \frac{|A|^n}{n} \cdot (\log c + n \cdot \log |A| - \log n) \\ &< c \cdot \frac{|A|^n}{n} \cdot n \cdot \log |A| = \frac{\log |A|}{k - 1 + \varepsilon} \cdot |A|^n , \end{aligned}$$

contradiction. Therefore $s(f) = S \geq c \cdot |A|^n / n$. □

Corollary 121. *Let* $\mathbf{A}$ *be a functionally complete ring or a functionally complete group or one of the two-element algebras* $\mathbf{B}$ *or* $\mathbf{B}_0$*. For every* $\varepsilon > 0$ *and for sufficiently large* n *(depending on* ε*) there exists an* n*-ary function* f_1 *over* $\mathbf{A}$ *such that*

$$s(f_1) \geq \frac{1}{1 + \varepsilon} \cdot \frac{|A|^n}{n}.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *(depending on* ε*) there exists an* n*-ary function* f_2 *over* $\mathbf{A}$ *such that*

$$d(f_2) \geq n \cdot \log |A| - \log \log n + \log \log |A| - \log (1 + \varepsilon).$$

Proof. We apply Theorem 120 with $k = 2$. □

We summarize our bounds for some two-element functionally complete algebras.

Corollary 122. *Let* $\mathbf{A}$ *be one of the two-element algebras* $\mathbf{B}$, $\mathbf{B}_0$ *or* $\mathbf{Z}_2$. *For an arbitrary* n*-ary function* f *over* $\mathbf{A}$ *we have*

$$s(f) \leq 6 \cdot (2^n - 1),$$
$$d(f) \leq n + \lceil \log n \rceil + 1.$$

For every $\varepsilon > 0$ *and for sufficiently large* n *(depending on* ε*) there exists an* n*-ary function* f_1 *over* $\mathbf{A}$ *such that*

$$s(f_1) \geq \frac{1}{1+\varepsilon} \cdot \frac{|2|^n}{n}.$$

Moreover for every $\varepsilon > 0$ *and for sufficiently large* n *(depending on* ε*) there exists an* n*-ary function* f_2 *over* $\mathbf{A}$ *such that*

$$d(f_2) \geq n - \log\log n - \log(1+\varepsilon).$$

Proof. We apply Corollaries 119 and 121. □

Remark 123. Lupanov [24] considered the algebra $\mathbf{A}$ over $\{0,1\}$ which contains all 16 binary operations as basic operations. He proved that for an arbitrary n-ary function f over $\{0,1\}$ we have $s(f)_{\mathbf{A}} \leq (1+o(1)) \cdot 2^n/n$. Gaskov [8] proved that for an arbitrary n-ary function f over $\{0,1\}$ we have $d(f)_{\mathbf{A}} \leq n - \log\log n + 2 + o(1)$.

The definition of size and depth of a function is robust in the sense that a complexity of a function over different functionally complete algebras differs only by a constant factor depending on the algebras:

Proposition 124. *Let* $\mathbf{A}_1$ *and* $\mathbf{A}_2$ *be two functionally complete algebras with underlying sets* A_1 *and* A_2. *Let* $e\colon A_1 \hookrightarrow A_2^l$ *be an embedding of* A_1 *to* A_2^l *for some* l. *For every* m, *let* $e^m\colon A_1^m \hookrightarrow A_2^{l \cdot m}$ *be the* m*th power of the embedding* e *and let* $(e^m)^{-1}$ *be the partial inverse of* e^m. *Let* $f\colon A_1^n \to A_1^k$ *be an arbitrary (possibly partial) function.*

$$\begin{array}{ccc} A_1^n & \xrightarrow{\quad f \quad} & A_1^k \\ \big\downarrow{\scriptstyle e^n} & & \big\downarrow{\scriptstyle e^k} \\ A_2^{n \cdot l} & \xrightarrow[e^k \circ f \circ (e^n)^{-1}]{} & A_2^{k \cdot l} \end{array}$$

Then there exist constants $c_s = c_s(\mathbf{A}_1, \mathbf{A}_2, e), c_d = c_d(\mathbf{A}_1, \mathbf{A}_2, e)$ *such that*

$$s_{\mathbf{A}_2}\left(e^k \circ f \circ (e^n)^{-1}\right) \leq c_s \cdot s_{\mathbf{A}_1}(f),$$
$$d_{\mathbf{A}_2}\left(e^k \circ f \circ (e^n)^{-1}\right) \leq c_d \cdot d_{\mathbf{A}_1}(f).$$

Proof. The idea of the proof is to compute the basic functions of $\mathbf{A}_1$ with circuits over $\mathbf{A}_2$. Then replace by these circuits every gate in the circuit computing the function f. We prove the inequality for the size, the same argument works for the depth.

Let the basic operations of $\mathbf{A}_1$ be $g_1, \ldots, g_m$ with arity $n_1, \ldots, n_m$. Now let $g_i' = e \circ g_i \circ (e^{n_i})^{-1} : A_2^{l \cdot n_i} \to A_2^l$ and $\mathbf{A}_2$ is functionally complete, therefore g_i' can be computed by a $\mathbf{A}_2$-circuit C_i. We can assume without loss of generality that $s_{\mathbf{A}_2}(g_i') = s(C_i)$. Now let

$$c_s = \max_{1 \leq i \leq m} s(C_i) = \max_{1 \leq i \leq m} s_{\mathbf{A}_2}(g_i').$$

Let C_s be an $\mathbf{A}_1$-circuit computing f, such that $s(C_s) = s_{\mathbf{A}_1}(f)$. Now we replace in C_s every gate, labelled by g_i (for every $1 \leq i \leq m$), by its corresponding circuit C_i. Moreover, we replace the variable x_j (for $1 \leq j \leq n$) by the variables $x_{j,1}, \ldots, x_{j,l}$. The circuit we obtain computes $e^k \circ f \circ (e^n)^{-1}$ and has size at most $c_s \cdot s(C_s) = c_s \cdot s_{\mathbf{A}_1}(f)$. □

This proposition shows that whenever we want to compute functions over different functionally complete algebras, we only have to compute the basic operations of one algebra using the other algebra, and we can then derive upper bounds on the complexities. In the following Section we find circuits computing an arbitrary function over a functionally complete group using the ideas of Section 2.3. Then we compare the functionally complete groups (especially alternating groups) to other functionally complete algebras in the terms of circuit complexity. We investigate especially the case where the other algebra is a field of prime order or one of the two-element algebras $\mathbf{B}$ and $\mathbf{B}_0$.

4.2 Functionally complete groups

In this Section we consider functionally complete groups $\mathbf{G}$ from the circuit complexity perspective. For an arbitrary n-ary function $f : G^n \to G$ we build a circuit which computes f. Then we give upper bounds on the size and on the depth of the constructed circuit (we gave lower bounds in Corollary 121). Let us start with some easy observations.

Proposition 125. *Let $\mathbf{G}$ be a functionally complete group, let f be an arbitrary n-ary (possibly partial) function over $\mathbf{G}$. Then*

$$\begin{gathered}\lceil \log \|f\| \rceil = d^*(f) \leq s^*(f) \leq \|f\| - 1, \\ s^*(f) \leq s(f) \leq \|f\| + n - 1, \\ \lceil \log \|f\| \rceil \leq d(f) \leq \lceil \log \|f\| \rceil + 1.\end{gathered}$$

Proof. Let p be a polynomial realizing f over $\mathbf{G}$ such that $\|f\| = \|p\|$. By Proposition 74 we can assume that every inverse in the polynomial p is used on variables. Let us consider a $\mathbf{G}$-circuit C_1 corresponding to the polynomial p. This circuit contains at most $\|p\| - 1$-many non-unary gates, since p contains $\|p\| - 1$-many binary group multiplications, therefore $s^*(f) \leq s^*(C_1) \leq \|f\| - 1$. As every inverse is used only on variables, we need to use at most n-many unary gates (labelled by the inverse operation), hence $s(f) \leq s^*(f) + n \leq \|f\| + n - 1$.

Moreover, by the associativity of the group multiplication, the $\|f\| - 1$-many multiplications can be executed in any order, not only as in the polynomial p. Let $l = \|p\|$ and let $p = w_1 w_2 \dots w_l$ (omitting the parentheses), where every w_i is a constant, or a variable, or an inverse of a variable. Then the following circuit C_2 has non-unary depth $\lceil \log \|f\| \rceil$: first execute every $w_{2i-1} \cdot w_{2i}$ for every $1 \leq i \leq l/2$ parallelly. Then execute every $(w_{2i-1}w_{2i}) \cdot (w_{2i+1}w_{2i+2})$ for every $1 \leq i \leq l/4$ parallelly, etc. Using this idea we do exactly $\lceil \log \|f\| \rceil$-many parallel multiplications, and so $d^*(f) \leq d^*(C_2) \leq \lceil \log \|f\| \rceil$. As every inverse is used only on variables, we have $d(f) \leq d^*(f) + 1 \leq \lceil \log \|f\| \rceil + 1$.

The remaining inequalities follow from Propositions 112 and 114. □

Remark 126. The connection between the depth and the length is certainly an important property of functionally complete groups. For every other algebra we are only able to give the logarithmic lower bound which might not be sharp. Proposition 125 shows that the trivial lower bound for depth can almost be achieved, moreover by a circuit which corresponds to a minimal length polynomial realization. It is open whether the length and the size can be minimized with the same circuit.

We remind the reader of some notations from Chapter 3. Let $\mathbf{G}$ be a functionally complete group, let $N = |\mathbf{G}|$. For every $1 \neq u \in G$ and for every $v \in G$ let $p_{u,v}$ be the unary partial function for which $p_{u,v}(1) = 1$ and $p_{u,v}(u) = v$. Let $f_b^{(n)}$ (for $b \neq 1$) be the n-ary partial function defined in Lemma 21, i.e. $f_b^{(n)}(b, \dots, b) = b$ and $f_b^{(n)}(x_1, \dots, x_n) = 1$ if $x_i = 1$ for some $1 \leq i \leq n$. Let $\chi_{1;u}$ (for $u \neq 1$) be the unary characteristic function described in Lemma 23, i.e. $\chi_{1;u}(1) = u$ and $\chi_{1;u}(x) = 1$ if $x \neq 1$. Finally let $\chi_{a_1,\dots,a_n;u}$ be the n-ary characteristic function described in Lemma 25, i.e. $\chi_{a_1,\dots,a_n;u}(a_1, \dots, a_n) = u$ and $\chi_{a_1,\dots,a_n;u}(x_1, \dots, x_n) = 1$, whenever $x_i \neq a_i$ for some i.

Let $V = v\left(f_b^{(2)}\right)$. For every $1 \neq u \in G$, for every $v \in G$, and for every

subset $S \subseteq G$ let

$$\begin{aligned} K_{u,v} &= v\left(p_{u,v}\right), \\ K_{S,v} &= \max\left\{ K_{u,v} : 1 \neq u \in S \right\}, \\ K_{u,S} &= \max\left\{ K_{u,v} : v \in S \right\}. \end{aligned}$$

Let $K = 1 + \max\left\{ K_{G\setminus\{1\},b}, K_{b,G\setminus\{1\}} \right\}$. We note here that K is bounded by the number of conjugacy classes of $\mathbf{G}$ by Proposition 79. Using Proposition 125 we can give an upper bound on the depth:

Theorem 127. *Let $\mathbf{G}$ be a functionally complete group. Let f be an arbitrary n-ary (possibly partial) function over $\mathbf{G}$ with e-many non-identity values ($e \geq 1$). Then the following inequalities hold:*

$$\begin{aligned} d\left(f\right) &\leq 2 + \log K_{G\setminus\{1\},b} + \log K_{b,G\setminus\{1\}} + \log V \cdot \left(2 + \log\left(N-1\right) + \log n\right) \\ &\quad + \log e, \\ d\left(f\right) &\leq 14 + 2\log\left(K-1\right) + 8\log\left(N-1\right) + 8\log n + \log e. \end{aligned}$$

If $\mathbf{G} = \mathbf{A}_m$ ($m \geq 5$), then

$$d\left(f\right) \leq 1 + \log m + 2 \cdot \left(\log 3 + \log N + \log n\right) + \log e.$$

If $4 \nmid m$, then the constant 1 at the beginning of the formula can be omitted.

Proof. We apply Theorems 75, 88 and Proposition 125. □

The following theorem gives upper bounds on the size of several (possibly partial) functions over $\mathbf{G}$.

Theorem 128. *Let $\mathbf{G}$ be a functionally complete group. Let f be an n-ary (possibly partial) function over $\mathbf{G}$ with e-many non-identity values. Let $N = |\mathbf{G}|$ and let $K = 1 + \max\left\{ K_{G\setminus\{1\},b}, K_{b,G\setminus\{1\}} \right\}$. Then K is at most the number of conjugacy classes in $\mathbf{G}$ and*

$$s\left(p_{u,v}\right) \leq 2 \cdot K_{u,v} + 1, \tag{4.11}$$

$$s\left(f_b^{(n)}\right) \leq 6 \cdot n - 6 + \max_{u \neq 1} s\left(p_{u,b}\right), \tag{4.12}$$

$$s\left(\chi_{1;b}\right) \leq s\left(f_b^{(N-1)}\right) + \sum_{u \neq 1} \left(2 + s\left(p_{u,b}\right)\right), \tag{4.13}$$

$$s\left(\chi_{a_1,\ldots,a_n;b}\right) \leq s\left(f_b^{(n)}\right) + n \cdot \left(1 + s\left(\chi_{1;b}\right)\right), \tag{4.14}$$

$$s\left(\chi_{a_1,\ldots,a_n;u}\right) \leq s\left(\chi_{a_1,\ldots,a_n;b}\right) + s\left(p_{b,u}\right), \tag{4.15}$$

$$s\left(f\right) \leq e \cdot \left(1 + \max_{u \neq 1} s\left(\chi_{a_1,\ldots,a_n;u}\right)\right) - 1, \tag{4.16}$$

$$s(f) \leq e \cdot \left(9nN - 7n - 3 + 2K_{b,G\setminus\{1\}} + 2(nN+1) K_{G\setminus\{1\},b}\right) - 1, \quad (4.17)$$
$$s(f) \leq e \cdot (9nN \cdot (2K+7) - 7n - 7 + 4K) - 1. \quad (4.18)$$

Moreover if $\mathbf{G} = \mathbf{A}_m$ *(*$m \geq 5$*), then*

$$s(f) \leq e \cdot ((27N - 14) \cdot n + m - 2) - 1.$$

If $4 \nmid m$*, then we can replace the factor* $(27N - 14)$ *by* $(13N - 11)$ *and the factor* m *by* $2 \cdot \lfloor m/2 \rfloor$*.*

Proof. The inequality (4.11) follows from Propositions 74 and 125. For proving inequality (4.12) we introduce a series of elements u_n of G. Let $u_1 = b$, we define u_i inductively such that $u_i \neq 1$ for every i. By Lemma 20 there exists c_i such that $[u_{i-1}, b^{c_i}] \neq 1$. Choose c_i and let $u_i = [u_{i-1}, b^{c_i}] \neq 1$. Let $h_1(x_1) = x_1$ and for every k let $h_k(x_1, \ldots, x_k) = [h_{k-1}(x_1, \ldots, x_{k-1}), x_k^{c_k}]$. By Lemma 21 we know that $p_{u_n,b}(h_n(x_1, \ldots, x_n))$ is a good representation of $f_b^{(n)}$. Now it is easy to see by induction that $s(h_k) \leq 6n - 6$, as commutating can be done in size 4: calculate $x \cdot y$, $y \cdot x$, then $(y \cdot x)^{-1}$ and finally $(y \cdot x)^{-1} \cdot (x \cdot y)$. Using Proposition 115 we have inequality (4.12).

The inequalities (4.13), (4.14), (4.15), (4.16) follow from Proposition 115 using on the following representations based on the proof of Theorem 18:

$$\begin{aligned}
\chi_{1;b}(x) &= f_b^{(N-1)}\left(bp_{u_2,b}(x)^{-1}, \ldots, bp_{u_N,b}(x)^{-1}\right), \\
\chi_{a_1,\ldots,a_n;b}(x_1, \ldots, x_n) &= f_b^{(n)}\left(\chi_{1;b}\left(x_1 a_1^{-1}\right), \ldots, \chi_{1;b}\left(x_n a_n^{-1}\right)\right), \\
\chi_{a_1,\ldots,a_n;u}(x_1, \ldots, x_n) &= p_{b,u}\left(\chi_{a_1,\ldots,a_n;b}(x_1, \ldots, x_n)\right), \\
f(x_1 \ldots, x_n) &= \prod_{\substack{(a_1,\ldots,a_n)\in G^n \\ 1\neq u = f(a_1\ldots,a_n)}} \chi_{a_1,\ldots,a_n;u}(x_1, \ldots, x_n),
\end{aligned}$$

where $G = \{1, u_2, \ldots, u_N\}$. The inequality (4.17) follows from the former inequalities. Finally the inequality (4.18) follows from the inequality (4.17).

If $\mathbf{G} = \mathbf{A}_m$, then we can choose b as a 3-cycle. Now by Proposition 95 we have $K_{G\setminus\{1\},b} \leq 4$ and whenever $4 \nmid m$, then $K_{G\setminus\{1\},b} \leq 2$. By Proposition 97 we have $K_{b,G\setminus\{1\}} \leq \lfloor m/2 \rfloor$. Moreover by the proof of Proposition 90 it is easy to see that for every n we can represent $f_b^{(n)}(x_1, \ldots, x_n)$ with $\left[\left[\left[x_1^{c'_1}, x_2^{c'_2}\right], x_3^{c'_3}\right], \ldots, x_n^{c'_n}\right]$ for some constants $c'_1, \ldots, c'_n \in G$ (as we can choose the constants of h_n such that $h_n(b, \ldots, b)$ is a 3-cycle). From this representation we can conclude by induction that $s\left(f_b^{(n)}\right) \leq 6n - 4$. Now

applying the inequalities (4.13), (4.14), (4.15), (4.16) we have

$$\begin{aligned} s\left(\chi_{1;b}\right) &\leq 6N-10+11N-11 \leq 27N-21, \\ s\left(\chi_{a_1,\ldots,a_n;b}\right) &\leq 6n-4+n\cdot(27N-20) \leq (27N-14)\cdot n-4, \\ s\left(\chi_{a_1,\ldots,a_n;u}\right) &\leq (27N-14)\cdot n-4+2\cdot\lfloor m/2\rfloor+1 \leq (27N-14)\cdot n+m-3, \\ s\left(f\right) &\leq e\cdot((27N-14)\cdot n+m-2)-1. \end{aligned}$$

If $4 \nmid m$, then

$$\begin{aligned} s\left(\chi_{1;b}\right) &\leq 6N-10+7N-7 \leq 13N-17, \\ s\left(\chi_{a_1,\ldots,a_n;b}\right) &\leq 6n-4+n\cdot(13N-17) \leq (13N-11)\cdot n-4, \\ s\left(\chi_{a_1,\ldots,a_n;u}\right) &\leq (13N-11)\cdot n-4+2\cdot\lfloor m/2\rfloor+1 \\ &\leq (13N-11)\cdot n+2\cdot\lfloor m/2\rfloor-3, \\ s\left(f\right) &\leq e\cdot((13N-11)\cdot n+2\cdot\lfloor m/2\rfloor-2)-1. \end{aligned}$$

□

Remark 129. We have to observe that the representations used in the proof of Theorem 128 do not minimize the depth, e.g. $d\left(\left[\left[\left[x_1^{c'_1},x_2^{c'_2}\right],x_3^{c'_3}\right],\ldots,x_n^{c'_n}\right]\right)=3n-1$, but using Proposition 125 on $\left\|\left[\left[\left[x_1^{c'_1},x_2^{c'_2}\right],x_3^{c'_3}\right],\ldots,x_n^{c'_n}\right]\right\|=3\cdot 2^n-3$ we have $d\left(\left[\left[\left[x_1^{c'_1},x_2^{c'_2}\right],x_3^{c'_3}\right],\ldots,x_n^{c'_n}\right]\right)\leq n+1+\log 3$. Generally it is not possible to minimize the size and the depth with the same circuit.

4.3 Comparison with two-element algebras

In this Section we are going to compare functionally complete groups with two-element algebras. Algebras over the set $\{0,1\}$ have the most importance in Computer Science as computers are based on them. In particular, computers are based on the algebra $\mathbf{B}_0=(\{0,1\},\mathrm{NAND},\mathrm{NOR})$. In the theory of Boolean functions another algebra is investigated as well: the algebra with underlying set $\{0,1\}$ which has all binary operations over $\{0,1\}$ as basic operations. Beside these algebras we investigate the two-element Boolean algebra $\mathbf{B}=(\{0,1\},\neg,\wedge,\vee)$ and the two-element field $\mathbf{Z}_2=(\{0,1\},+,\cdot)$.

We are interested about the possible efficiency of functionally complete groups when computing different functions by circuits. By Proposition 124 we know that one functionally complete algebra can be more efficient than another by only a constant factor. Moreover, this constant factor is determined by only simulating the basic operations. Therefore if we want to know how much faster or slower functionally complete groups can be than algebras

over $\{0,1\}$, we have to simulate one's basic operations with the other. In this Section we simulate every binary function over $\{0,1\}$ with the group operations of a functionally complete group.

There are 16 binary functions over $\{0,1\}$. Two of them are the constant 0 and 1 function, four of them are unary (namely x, y, $\neg x = 1-x$, $\neq y = 1-y$) and 10 of them depending on both variables. These functions are $x \wedge y = x \cdot y$, $x \vee y$, $x + y$, $\neg x \wedge y$, $x \wedge \neg y$ and their negations.

In order to build a $\mathbf{G}$-circuit for computing these functions, we need an embedding $\{0,1\} \hookrightarrow G$. We assign the identity element $1 \in G$ of the group for $0 \in \mathbf{B}$ and we assign an element $1 \neq b \in G$ of the group for $1 \in \mathbf{B}$. As the function $f_b^{(2)}$ plays an important role in the simulation of binary $\{0,1\}$-functions, we choose b such that $s\left(f_b^{(2)}\right)$ or $d\left(f_b^{(2)}\right)$ is minimal. Moreover let $1 \neq u \in G$ be an element of order two. If $b^2 = 1$ then let $u = b$. Let $S = s\left(f_b^{(2)}\right)$, $D = d\left(f_b^{(2)}\right)$, $S_1 = s(p_{b,u})$, $D_1 = d(p_{b,u})$, $S_2 = s(p_{u,b})$, $D_2 = d(p_{u,b})$.

Table 4.1 shows a representation of the binary functions. Moreover, it contains trivial upper bounds on the size and the depth of these representations.

The following theorem compares the circuit complexity and depth of a function for two-element algebras with the circuit complexity and depth for functionally complete groups.

Theorem 130. *Let $\mathbf{G}$ be a functionally complete group and let K be its number of conjugacy classes. Let $\mathbf{A}$ denote the algebra with underlying set $\{0,1\}$ which has all binary operations over $\{0,1\}$ as basic operations. Let $\mathbf{B} = (\{0,1\}, \neg, \wedge, \vee)$ be the two-element Boolean algebra, let $\mathbf{B}_0 = (\{0,1\}, \mathrm{NAND}, \mathrm{NOR})$, and let $\mathbf{Z}_2 = (\{0,1\}, +, \cdot)$ the two-element field. Then there exists $1 \neq b \in G$ such that for every positive integer n and any function $f\colon \{0,1\}^n \to \{0,1\}$ we can find functions p_1, p_2 over $\mathbf{G}$ such that p_1 and p_2 are the same function over $\{1,b\}$ as f is over $\{0,1\}$ and*

$$
\begin{aligned}
&s_{\mathbf{G}}(p_1) \leq (6K + 456) \cdot s_{\mathbf{A}}(f), && d_{\mathbf{G}}(p_2) \leq (14 + 2\log K) \cdot d_{\mathbf{A}}(f),\\
&s_{\mathbf{G}}(p_1) \leq 456 \cdot s_{\mathbf{B}}(f), && d_{\mathbf{G}}(p_2) \leq 14 \cdot d_{\mathbf{B}}(f),\\
&s_{\mathbf{G}}(p_1) \leq 454 \cdot s_{\mathbf{B}_0}(f), && d_{\mathbf{G}}(p_2) \leq 12 \cdot d_{\mathbf{B}_0}(f),\\
&s_{\mathbf{G}}(p_1) \leq (6K + 448) \cdot s_{\mathbf{Z}_2}(f), && d_{\mathbf{G}}(p_2) \leq (10 + 2\log K) \cdot d_{\mathbf{Z}_2}(f).
\end{aligned}
$$

If $\mathbf{G} = \mathbf{A}_m$ (for $m \geq 5$) and $b = (1\,2\,3)$, then for every positive integer number n and any function $f\colon \{0,1\}^n \to \{0,1\}$ we can find functions p_1, p_2 over $\mathbf{G}$ such that p_1 and p_2 are the same function over $\{1,b\}$ as f is over

Table 4.1: Simulating binary functions over $\{0,1\}$

f over $\{0,1\}$	p over $\mathbf{G}$	$s(p)$	$d(p)$
0	1	0	0
1	b	0	0
x	x	0	0
y	y	0	0
$\neg x$	$b \cdot x^{-1}$	2	2
$\neg y$	$b \cdot y^{-1}$	2	2
$x \cdot y = x \wedge y$	$f_b^{(2)}(x,y)$	S	D
$\neg x \wedge y$	$f_b^{(2)}(bx^{-1},y)$	$2+S$	$2+D$
$x \wedge \neg y$	$f_b^{(2)}(x,by^{-1})$	$2+S$	$2+D$
$x \vee y$	$b \cdot \left(f_b^{(2)}(bx^{-1},by^{-1})\right)^{-1}$	$6+S$	$4+D$
$x+y$	$p_{u,b}(p_{b,u}(x) \cdot p_{b,u}(y))$	$1+2S_1+S_2$	$1+S_1+S_2$
$\neg(x \wedge y)$	$b \cdot \left(f_b^{(2)}(x,y)\right)^{-1}$	$2+S$	$2+D$
$\neg(\neg x \wedge y)$	$b \cdot \left(f_b^{(2)}(bx^{-1},y)\right)^{-1}$	$4+S$	$4+D$
$\neg(x \wedge \neg y)$	$b \cdot \left(f_b^{(2)}(x,by^{-1})\right)^{-1}$	$4+S$	$4+D$
$\neg(x \vee y)$	$f_b^{(2)}(bx^{-1},by^{-1})$	$4+S$	$2+D$
$1-x+y$	$b \cdot (p_{u,b}(p_{b,u}(x) \cdot p_{b,u}(y)))^{-1}$	$3+2S_1+S_2$	$3+S_1+S_2$

$\{0,1\}$ *and*

$$
\begin{aligned}
s_{\mathbf{A}_m}(p_1) &\leq 13 \cdot s_{\mathbf{A}}(f), & d_{\mathbf{A}_m}(p_2) &\leq 8 \cdot d_{\mathbf{A}}(f),\\
s_{\mathbf{A}_m}(p_1) &\leq 10 \cdot s_{\mathbf{B}}(f), & d_{\mathbf{A}_m}(p_2) &\leq 5 \cdot d_{\mathbf{B}}(f),\\
s_{\mathbf{A}_m}(p_1) &\leq 10 \cdot s_{\mathbf{B}_0}(f), & d_{\mathbf{A}_m}(p_2) &\leq 5 \cdot d_{\mathbf{B}_0}(f),\\
s_{\mathbf{A}_m}(p_1) &\leq 11 \cdot s_{\mathbf{Z}_2}(f), & d_{\mathbf{A}_m}(p_2) &\leq 6 \cdot d_{\mathbf{Z}_2}(f).
\end{aligned}
$$

If $\mathbf{G} = \mathbf{A}_m$ *for* $m \geq 6$ *then we can choose* $b = (1\,2)(3\,4)$ *and we can replace the constants* 13, 11, 8 *and* 6 *by* 10, 10, 5 *and* 5, *respectively.*

Proof. We use the representations and upper bounds given in Table 4.1. Applying Propositions 86, 74 and 125 we obtain $S \leq 450$ and $D \leq 10$. By Propositions 79 and 125 we obtain $S_1 \leq 2K-1$, $S_2 \leq 2K-1$, $D_1 \leq 2+\log K$, $D_2 \leq 2+\log K$ for an arbitrary element $1 \neq u \in G$ with $u^2 = 1$. Applying Table 4.1 we have the desired inequalities.

If $\mathbf{G} = \mathbf{A}_m$, then we can choose $b = (1\,2\,3)$ and let $u = (1\,3)\,(2\,4)$. Then we know by Propositions 90 and 125 that $S \leq 10$ and $D \leq 5$. Moreover it is easy to see that not only $f_b^{(2)}(x, y)$ has length 9 but every polynomial in Table 4.1 which involves $f_b^{(2)}$ has length 9 as well. We have $S_1 \leq 3$ and $D_1 \leq 2$ by having $p_{b,u}(x) = x \cdot c^{-1} \cdot x \cdot c$ with $c = (3\,4\,5)$. As we have $p_{u,b}(x) = c_1 \cdot x \cdot c_2 \cdot x \cdot c_3$ with $c_1 = (1\,3)\,(2\,5)$, $c_2 = (1\,3\,2)$, $c_3 = (2\,5\,3)$, we obtain $S_2 \leq 4$ and $D_2 \leq 3$.

Finally if $\mathbf{G} = \mathbf{A}_m$ for $m \geq 6$ then we can choose $u = b = (1\,2)\,(3\,4)$, having $S_1 = D_1 = S_2 = D_2 = 0$ and $S = 10$, $D = 5$. □

As we see, we can simulate 2-element algebras quite efficiently with $\mathbf{A}_m$ for $m \geq 6$, as the two-element algebra can be at most 10 times faster by using a single processor and 5 times faster using multiple processors. The case where $m = 5$ and we simulate with $\mathbf{A}_5$ can be interesting, as the symmetry group of the icosahedron is $\mathbf{A}_5$. Therefore if a machine which is based on the symmetry states of an icosahedron will ever be built, then that machine will be based on the group $\mathbf{A}_5$.

We finish the Section with a lower bound on the efficiency of $\mathbf{G}$-circuits.

Theorem 131. *Let $\mathbf{G}$ be a functionally complete group and let $\mathbf{A}$ be a functionally complete algebra over $\{0, 1\}$ with at most binary basic operations. For every $1 \neq g \in \mathbf{G}$ let $\delta(g)$ be the maximal order of any subgroup of $\mathbf{G}$ not containing g and let $\delta(\mathbf{G}) = \min\{\delta(g) : 1 \neq g \in \mathbf{G}\}$. Let $e\colon G \hookrightarrow \{0, 1\}^l$ be an embedding. Let us suppose that $f\colon \{0, 1\}^{2l} \to \{0, 1\}^l$ is a function such that $f(e(x), e(y)) = e(x \cdot y)$. Then*

$$s_{\mathbf{A}}(f) \geq \lceil \log |\mathbf{G}| \rceil,$$
$$d_{\mathbf{A}}(f) \geq 1 + \left\lceil \log\log \frac{|\mathbf{G}|}{\delta(\mathbf{G})} \right\rceil.$$

Proof. The first inequality is quite clear. First, $l \geq \lceil \log |\mathbf{G}| \rceil$, otherwise e cannot be an embedding. Since $\mathbf{A}$ has only binary basic operations and f has to depend on at least $\lceil \log |\mathbf{G}| \rceil$-many variables, we obtain $s_{\mathbf{A}}(f) \geq \lceil \log |\mathbf{G}| \rceil$.

The second inequality follows from a result of Spira [35]. He derives the lower bound $1 + \left\lceil \log\log \frac{|\mathbf{G}|}{\delta(\mathbf{G})} \right\rceil$ for the required time for realizing the $\mathbf{G}$-multiplication by a logical circuit. We have to observe that Spira's model (which is the same as e.g. Winograd's model in [43] and [44]) is quite similar as our circuit model, although he allows the circuits to contain cycles. In particular the required time in Spira's model is the same as the depth in our circuit model. □

4.4 Simulating rings by groups

In this Section first we build a $\mathbf{G}$-circuit which simulates an arbitrary ring $\mathbf{R}$. This simulation is rather 'brute force', Theorem 132 gives the details. It basically compares the sizes and the depths of $\mathbf{R}$-circuits and $\mathbf{G}$-circuits computing the same functions.

Then we introduce another method by which we can simulate the ring $\mathbf{Z}_p$ for an odd prime p. For every ring-polynomial q we build a $\mathbf{A}_m$-circuit (for $m \geq p+2$), which has linear size in $\|q\|$. Whenever for some constant c we have $s_{\mathbf{Z}_p}(f) \leq c \cdot \|f\|_{\mathbf{Z}_p}$ or $d_{\mathbf{Z}_p}(f) \leq c \cdot \|f\|_{\mathbf{Z}_p}$, then we can compute f by an $\mathbf{A}_m$-circuit C, such that $s(C)$ is linear in $s_{\mathbf{Z}_p}(f)$ or $d(C)$ is linear in $d_{\mathbf{Z}_p}(f)$.

Let us start first with the comparison of $\mathbf{R}$-circuits and $\mathbf{G}$-circuits.

Theorem 132. *Let $\mathbf{G}$ be a functionally complete group, let K be its number of conjugacy classes, and let $N = |\mathbf{G}|$. Let $\mathbf{R}$ be a finite ring. Let $l = \lceil \log_{|\mathbf{G}|} \mathbf{R} \rceil$ and let $e\colon \mathbf{R} \hookrightarrow \mathbf{G}^l$ be an embedding. Then for any n-ary function $f\colon \mathbf{R}^n \to \mathbf{R}$ which can be represented by an $\mathbf{R}$-polynomial we can find functions p_1, p_2 over $\mathbf{G}$ such that*

$$p_1(e(x_1), \ldots, e(x_n)) = e(f(x_1, \ldots, x_n)) = p_2(e(x_1), \ldots, e(x_n))$$

and

$$\begin{aligned}
&s_{\mathbf{G}}(p_1) \leq (9lN+1) \cdot (4K+14) \cdot N^{2l} \cdot s_{\mathbf{R}}(f),\\
&s_{\mathbf{G}}(p_1) \leq (9lN+1) \cdot (4K+14) \cdot N^2 \cdot |\mathbf{R}|^2 \cdot s_{\mathbf{R}}(f),\\
&d_{\mathbf{G}}(p_2) \leq (14 + 2\log K + 8\log N + 8 + 8\log l + 2l\log N) \cdot d_{\mathbf{R}}(f),\\
&d_{\mathbf{G}}(p_2) \leq (14 + 2\log K + 8\log N + 8 + 8\log l + 2\log N + 2\log|\mathbf{R}|) \cdot d_{\mathbf{R}}(f).
\end{aligned}$$

If $\mathbf{G} = \mathbf{A}_m$ (for $m \geq 5$), then for any n-ary function $f\colon \mathbf{R}^n \to \mathbf{R}$ we can find functions p_1, p_2 over $\mathbf{G}$ such that

$$p_1(e(x_1), \ldots, e(x_n)) = e(f(x_1, \ldots, x_n)) = p_2(e(x_1), \ldots, e(x_n))$$

and

$$\begin{aligned}
&s_{\mathbf{A}_m}(p_1) \leq (2l \cdot (27N-14) + m) \cdot N^{2l} \cdot s_{\mathbf{R}}(f),\\
&s_{\mathbf{A}_m}(p_1) \leq (2l \cdot (27N-14) + m) \cdot N^2 \cdot |\mathbf{A}_m|^2 \cdot s_{\mathbf{R}}(f),\\
&d_{\mathbf{A}_m}(p_2) \leq (3 + 2\log 3 + \log m + 2\log l + 2l \cdot \log N) \cdot d_{\mathbf{R}}(f),\\
&d_{\mathbf{A}_m}(p_2) \leq (3 + 2\log 3 + \log m + 2\log l + 2 \cdot \log N + 2\log|\mathbf{R}|) \cdot d_{\mathbf{R}}(f).
\end{aligned}$$

If $4 \nmid m$, then we can replace the factor $(27N-14)$ by $(13N-11)$ and the factor m by $2 \cdot \lfloor m/2 \rfloor$. in the bounds on $s_{\mathbf{A}_m}(p_1)$.

Proof. By Proposition 124 we only have to build a circuit for the ring addition and the ring multiplication. These are $2l$-ary partial functions over $\mathbf{G}$, therefore applying Theorems 127 and 128 gives us the desired bounds. □

Remark 133. We note that whenever $|\mathbf{R}| \leq |\mathbf{G}|$, then we can embed $\mathbf{R}$ into $\mathbf{G}$. Let S be the image of $\mathbf{R}$. Then we can consider the ring addition and ring multiplication as partial binary functions over S, and N can be replaced by $|S| = |\mathbf{R}|$ in the bounds of Theorem 132.

The following theorem gives us a lower bound on the efficiency of $\mathbf{G}$-circuits.

Theorem 134. *Let $\mathbf{G}$ be a functionally complete group and let $\mathbf{R}$ be a functionally complete ring. For every $1 \neq g \in \mathbf{G}$ let $\delta(g)$ be the maximal order of any subgroup of $\mathbf{G}$ not containing g and let $\delta(\mathbf{G}) = \min\{\delta(g) : 1 \neq g \in \mathbf{G}\}$. Let $e\colon G \hookrightarrow \mathbf{R}^l$ be an embedding. Let us suppose that $f\colon \mathbf{R}^{2l} \to \mathbf{R}^l$ is a function such that $f(e(x), e(y)) = e(x \cdot y)$, where $\cdot$ denotes the group multiplication in $\mathbf{G}$. Then*

$$s_{\mathbf{R}}(f) \geq \left\lceil \log_{|\mathbf{R}|} |\mathbf{G}| \right\rceil,$$
$$d_{\mathbf{R}}(f) \geq 1 + \left\lceil \log\log_{|\mathbf{R}|} \frac{|\mathbf{G}|}{\delta(\mathbf{G})} \right\rceil.$$

Proof. The first inequality is quite clear. First, $l \geq \left\lceil \log_{|\mathbf{R}|} |\mathbf{G}| \right\rceil$, otherwise e cannot be an embedding. Since $\mathbf{R}$ has only binary basic operations and f has to depend on at least $\left\lceil \log_{|\mathbf{R}|} |\mathbf{G}| \right\rceil$-many variables, we obtain $s_{\mathbf{A}}(f) \geq \left\lceil \log_{|\mathbf{R}|} |\mathbf{G}| \right\rceil$.

The second inequality follows from a result of Spira [35]. He derives the lower bound $1 + \left\lceil \log\log_{|\mathbf{R}|} \frac{|\mathbf{G}|}{\delta(\mathbf{G})} \right\rceil$ for the required time for realizing the $\mathbf{G}$-multiplication by a circuit. We have to observe that Spira's model (which is the same as e.g. Winograd's model in [43] and [44]) is quite similar as our circuit model, although he allows the circuits to contain cycles. In particular the required time in Spira's model is the same as the depth in our circuit model. □

In the following part of the Section we show another method which can be useful for simulating the ring $\mathbf{Z}_p$ with the alternating group $\mathbf{A}_m$ for $m \geq p+2$. Let $\mathbf{G} = \mathbf{A}_m$ be such an alternating group. Let $a = (1, 2, \ldots, p)$ and let $\mathbf{A} = \langle a \rangle$ an Abelian subgroup of $\mathbf{G}$. Let r be a primitive root modulo p. The elements a and a^r have the same cycle structure, therefore there exists an element $h' \in \mathbf{S}_p$ such that $a^{h'} = a^r$. If h' is even, then let $h = h' \in \mathbf{A}_m$, otherwise let $h = h' \cdot (p+1, p+2) \in \mathbf{A}_m$. Let $\mathbf{H} = \langle h \rangle$. The subgroup $\mathbf{H} \leq \mathbf{G}$

acts on $\mathbf{A}$ by conjugation and the action is isomorphic to $\mathbf{B} = \mathbf{H}/C_{\mathbf{H}}(\mathbf{A})$. Let $\varphi\colon \mathbf{H} \to \mathbf{B}$ be the natural homomorphism. Every element of $\mathbf{B}$ acts as an automorphism of $\mathbf{B}$, in particular every element is an endomorphism. Since $\mathbf{B}$ is commutative, the actions of $\mathbf{B}$ generate a finite nontrivial commutative subring $\mathbf{R}(\mathbf{B})$ of End $\mathbf{A} = \mathbf{Z}_p$. Let $b = \varphi(h)$, then $\mathbf{B} = \langle b \rangle$.

Now for any natural number t we have $a^{(b^t)} = a^{(h^t)} = a^{(r^t)}$. Since r is a primitive root modulo p, the elements b and h are of order $p-1$, therefore $|\mathbf{B}| = p-1$. Since $\mathbf{B} \cup \{0\} \subseteq \mathbf{R}(\mathbf{B}) \subseteq$ End $\mathbf{A} = \mathbf{Z}_p$ and $p = |\mathbf{B}| + 1 \leq |\mathbf{R}(\mathbf{B})| \leq |\text{End } \mathbf{A}| = p$, we have $|\mathbf{R}(\mathbf{B})| = p$ and $\mathbf{R}(\mathbf{B}) = \mathbf{B} \cup \{0\} =$ End $\mathbf{A} = \mathbf{Z}_p$.

The idea is the following: for every $\mathbf{Z}_p$-polynomial $q(z_1, \ldots, z_n)$ we build a $\mathbf{G}$-circuit $C(q)$, which computes $a^{q(z_1,\ldots,z_n)}$ over $\mathbf{G}$, where $x^{y+z} = x^y x^z = y^{-1}xyz^{-1}xz$, $x^{-y} = (x^{-1})^y = y^{-1}x^{-1}y$ and $x^{yz} = (x^y)^z = (yz)^{-1}xyz$. Now let us consider the inputs $z_1, \ldots, z_n$ as elements of $\mathbf{Z}_p = \mathbf{R}(\mathbf{B}) =$ End $\mathbf{A}$. Then the circuit $C(q)$, for a suitable encoding of the inputs $z_1, \ldots, z_n$, computes $a^{q(z_1,\ldots,z_n)} \in \mathbf{A}$. Now we read the result of the computation as an element of $\mathbf{Z}_p$ considering $\mathbf{A} \simeq (\mathbf{Z}_p, +)$. This idea can be applied for simulating more general finite rings.

There is a slight problem with this construction, therefore some refinements are necessary. The input z_i can attain p-many values when we consider it as input for the $\mathbf{Z}_p$-polynomial q. On the other hand, when z_i is considered as an input of the circuit C, then it can only attain *automorphisms* as value from $\mathbf{R}(\mathbf{B})$. More precisely z_i attains values from the group $\mathbf{B}$, never from $\mathbf{R}(\mathbf{B}) \setminus \mathbf{B}$. On the other hand $\mathbf{B}$ generates $\mathbf{R}(\mathbf{B})$: the polynomial $y - y'$ has the property that if $y, y' \in \mathbf{B}$, then $y - y' \in \mathbf{R}(\mathbf{B})$ and for every $z \in \mathbf{R}(\mathbf{B})$ we can choose $y, y' \in \mathbf{B}$ such that $z = y - y'$. Therefore the above-mentioned idea works with substituting $z_i = y_i - y_i'$ in the polynomial q.

First we state a proposition which handles the situation when the polynomial q is 'nice'. Let us recall that by $v(q)$ we denoted the number of variable occurrences in the polynomial q.

Proposition 135. *Let $q'(z_1, \ldots, z_n)$ be a $\mathbf{Z}_p$-polynomial, which contains $add(q')$-many additions and does not contain subtraction or the constant 0. Then for $m \geq p+2$ there exists an $\mathbf{A}_m$-circuit $C(q')$ which computes the $\mathbf{A}_m$-function $a^{q'(z_1,\ldots,z_n)}$, where $a = (1, \ldots, p)$, $x^{y+z} = x^y x^z = y^{-1}xyz^{-1}xz$, $x^{-y} = (x^{-1})^y = y^{-1}x^{-1}y$, $x^{yz} = (x^y)^z = (yz)^{-1}xyz$ and*

$$s(C(q')) \leq add(q') + v_{\mathbf{Z}_p}(q') + 2\|q'\|_{\mathbf{Z}_p} \leq 4\|q'\|_{\mathbf{Z}_p},$$
$$d(C(q')) \leq 2\|q'\|_{\mathbf{Z}_p}.$$

Proof. We construct a circuit $C'(q')$ computing the function $x^{q'}$ by induction on q'. For a variable z let $C'(z)$ be a circuit which computes $z^{-1} \cdot x \cdot z$ in size 3 and in depth 2. Let r be a primitive root modulo p and let $h \in \mathbf{A}_m$ be an element for which $a^h = a^r$. Now every nonzero constant from $\mathbf{Z}_p$ is of the form r^k, represented by $a \mapsto a^{(r^k)}$ in End $\mathbf{A}$. Then for $0 \leq k \leq p-1$ let $C'\left(r^k\right)$ be a circuit which computes $\left(h^k\right)^{-1} \cdot x \cdot h^k$ in size 2 and in depth 2. Now let $q' = q'_1 + q'_2$. By induction we have circuits $C'(q'_1)$ and $C'(q'_2)$ computing $x^{q'_1}$ and $x^{q'_2}$ such that

$$\begin{aligned}
&s\left(C'\left(q'_1\right)\right) \leq add\left(q'_1\right) + v_{\mathbf{Z}_p}\left(q'_1\right) + 2\left\|q'_1\right\|_{\mathbf{Z}_p}, \\
&d\left(C'\left(q'_1\right)\right) \leq 2\left\|q'_1\right\|_{\mathbf{Z}_p}, \\
&s\left(C'\left(q'_2\right)\right) \leq add\left(q'_2\right) + v_{\mathbf{Z}_p}\left(q'_2\right) + 2\left\|q'_2\right\|_{\mathbf{Z}_p}, \\
&d\left(C'\left(q'_2\right)\right) \leq 2\left\|q'_2\right\|_{\mathbf{Z}_p}.
\end{aligned}$$

Now let $C'(q')$ be the circuit which contains both $C'(q'_1)$ and $C'(q'_2)$ parallelly, and multiplies the final gates of $C'(q'_1)$ and $C'(q'_2)$. Now $C'(q')$ clearly computes $x^{q'} = x^{q'_1} \cdot x^{q'_2}$. Using the bounds on the sizes and depths of $C'(q'_1)$ and $C'(q'_2)$, it is easy to see that

$$\begin{aligned}
s\left(C'\left(q'\right)\right) &\leq s\left(C'\left(q'_1\right)\right) + s\left(C'\left(q'_2\right)\right) + 1 \\
&\leq add\left(q'\right) + v_{\mathbf{Z}_p}\left(q'\right) + 2\left\|q'\right\|_{\mathbf{Z}_p}, \\
d\left(C'\left(q'\right)\right) &\leq 1 + \max\left\{ d\left(C'\left(q'_1\right)\right), d\left(C'\left(q'_2\right)\right) \right\} \leq d\left(C'\left(q'_1\right)\right) + d\left(C'\left(q'_2\right)\right) \\
&\leq 2\left\|q'\right\|_{\mathbf{Z}_p}.
\end{aligned}$$

The proof is very similar if $q' = q'_1 \cdot q'_2$. By induction we have circuits $C'(q'_1)$ and $C'(q'_2)$ computing $x^{q'_1}$ and $x^{q'_2}$ such that

$$\begin{aligned}
&s\left(C'\left(q'_1\right)\right) \leq add\left(q'_1\right) + v_{\mathbf{Z}_p}\left(q'_1\right) + 2\left\|q'_1\right\|_{\mathbf{Z}_p}, \\
&d\left(C'\left(q'_1\right)\right) \leq 2\left\|q'_1\right\|_{\mathbf{Z}_p}, \\
&s\left(C'\left(q'_2\right)\right) \leq add\left(q'_2\right) + v_{\mathbf{Z}_p}\left(q'_2\right) + 2\left\|q'_2\right\|_{\mathbf{Z}_p}, \\
&d\left(C'\left(q'_2\right)\right) \leq 2\left\|q'_2\right\|_{\mathbf{Z}_p}.
\end{aligned}$$

Now let $C'(q')$ be the circuit which contains both $C'(q'_1)$ and $C'(q'_2)$, but $C'(q'_2)$ is not applied on the variables $x, z_1, \ldots, z_n$, but on the final gate of $C'(q'_1)$ and on the variables $z_1, \ldots, z_n$. Now $C'(q')$ clearly computes $x^{q'} = \left(x^{q'_1}\right)^{q'_2}$. Using the bounds on the sizes and depths of $C'(q'_1)$ and $C'(q'_2)$, it

is easy to see that

$$\begin{aligned} s\left(C'\left(q'\right)\right) &\leq s\left(C'\left(q'_1\right)\right) + s\left(C'\left(q'_2\right)\right) \\ &\leq add\left(q'\right) + v_{\mathbf{Z}_p}\left(q'\right) + 2\left\|q'\right\|_{\mathbf{Z}_p}, \\ d\left(C'\left(q'\right)\right) &\leq d\left(C'\left(q'_1\right)\right) + d\left(C'\left(q'_2\right)\right) \\ &\leq 2\left\|q'\right\|_{\mathbf{Z}_p}. \end{aligned}$$

Finally we obtain $C\left(q'\right)$ from $C'\left(q'\right)$ by replacing every outgoing edge from x by an outgoing edge of a: if an edge was going from x to the gate G_i, then we remove it and add an edge from a to G_i. □

Now we can state the main theorem of this Section.

Theorem 136. *Let p be an odd prime and let $m \geq p+2$. Let $a = (1, \ldots, p) \in \mathbf{A}_m$, let r be a primitive root modulo p and let $h \in \mathbf{A}_m$ such that $a^h = a^r$. Let $\mathbf{H} = \langle h \rangle$ and let $\mathbf{A} = \langle a \rangle$. Let $in\colon \mathbf{Z}_p \hookrightarrow \mathbf{H} \times \mathbf{H}$ and $out\colon \mathbf{Z}_p \hookrightarrow \mathbf{A}$ be embeddings such that for every $0 \leq k \leq p-1$ we have $out\left(k\right) = a^k$ and $in\left(k\right) = \left(h^{k_1}, h^{k_2}\right)$ such that $r^{k_1} - r^{k_2} = k$ in $\mathbf{Z}_p$. Then for every $\mathbf{Z}_p$-polynomial $q\left(z_1, \ldots, z_n\right)$ there exists an $\mathbf{A}_m$-circuit C such that for every n-tuple $\left(r_1, \ldots, r_n\right)$ over $\mathbf{Z}_p$ the circuit C computes $out\left(q\left(r_1, \ldots, r_n\right)\right)$ on the input $2n$-tuple $\left(in\left(r_1\right), \ldots, in\left(r_n\right)\right)$ and*

$$\begin{aligned} s\left(C\right) &\leq 16\left\|q\right\|_{\mathbf{Z}_p}, \\ d\left(C\right) &\leq 8\left\|q\right\|_{\mathbf{Z}_p}. \end{aligned}$$

Proof. Let us replace in q every variable z_i by $y_i + (p-1) \cdot y'_i$, every constant 0 by $1 + (p-1)$, and every subtraction $q_0 - q_1$ by $q_0 + (p-1) \cdot q_1$ (for subpolynomials q_0 and q_1). Thus we obtain a polynomial q', such that $\left\|q'\right\| \leq 4 \cdot \left\|q\right\|$. Moreover for $z_i = y_i - y'_i$ $(1 \leq i \leq n)$ we have $q\left(z_1, \ldots, z_n\right) = q'\left(y_1, y'_1, \ldots, y_n, y'_n\right)$. By Proposition 135 we have a circuit C such that C computes $a^{q'\left(y_1, y'_1, \ldots, y_n, y'_n,\right)} = out\left(q'\left(y_1, y'_1, \ldots, y_n, y'_n,\right)\right)$ with

$$\begin{aligned} s\left(C\right) &\leq 4\left\|q'\right\|_{\mathbf{Z}_p} \leq 16\left\|q\right\|_{\mathbf{Z}_p}, \\ d\left(C\right) &\leq 2\left\|q'\right\|_{\mathbf{Z}_p} \leq 8\left\|q\right\|_{\mathbf{Z}_p}. \end{aligned}$$

□

The bounds on the size and on the depth in Theorem 136 show that whenever for some constant c we have $s_{\mathbf{Z}_p}\left(f\right) \leq c \cdot \left\|f\right\|_{\mathbf{Z}_p}$ or $d_{\mathbf{Z}_p}\left(f\right) \leq c \cdot \left\|f\right\|_{\mathbf{Z}_p}$, then we can compute f by an $\mathbf{A}_m$-circuit C, such that $s\left(C\right) \leq 16c \cdot s_{\mathbf{Z}_p}\left(f\right)$ or $d\left(C\right) \leq 8c \cdot d_{\mathbf{Z}_p}\left(f\right)$. Therefore this method of simulating the ring $\mathbf{Z}_p$ can be more efficient than that of Theorem 132 for certain functions.

4.5 Finite-state sequential circuits

In this Section we investigate a different approach for function realizations than that introduced in Section 4.1. Krohn, Maurer and Rhodes in [22] showed a method how finite-state sequential circuits can be used for calculating an arbitrary Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. They, however, did not measure the efficiency of their method. First, we recall their method, then we give an upper bound on the time required for calculating an arbitrary Boolean function $f\colon \{0,1\}^n \to \{0,1\}$.

A *finite-state sequential circuit* is a 6-tuple $\mathbf{M} = (A, B, Q, q_0, \lambda, \mu)$, with basic input set A, basic output set B, state set Q, starting state q_0, next-state function $\lambda\colon Q \times A \to Q$ and output function $\mu\colon Q \to B$. Let A^+ be the free semigroup generated by A, i.e. all finite words with positive length constructed from the alphabet A. For any $t = a_1 \cdots a_n \in A^+$ let us define $\lambda'(t) : Q \to Q$ inductively: $\lambda'(a_1)(q) = \lambda(q, a_1)$ for $a_1 \in A$ and $q \in Q$. Let $\lambda'(a_1 \cdots a_k)(q) = \lambda'(a_k)(\lambda'(a_1 \cdots a_{k-1})(q))$ for $a_1 \ldots a_k \in A^+$ and $q \in Q$. Let $\mathbf{M}_q(a_1 \ldots a_k) = \mu(\lambda'(a_1 \ldots a_k)(q))$. This is the letter which machine $\mathbf{M}$ when started in state q outputs for the word $a_1 \ldots a_k$.

Let $\mathbf{F}(Q)$ denote the semigroup of all transformations of Q into itself under the multiplication $\cdot$, where for $f, g \in \mathbf{F}(Q)$ we have $(f \cdot g)(q) = g(f(q))$. Then $\lambda'\colon A^+ \to \mathbf{F}(Q)$ is a homomorphism: $\lambda'(a_1 \ldots a_k b_1 \ldots b_m) = \lambda'(a_1 \ldots a_k) \cdot \lambda'(b_1 \ldots b_m)$. Let us denote $\lambda'(A^+)$ by $\mathbf{M}^S$. We call $\mathbf{M}^S$ the *semigroup of the machine* $\mathbf{M}$.

Definition 137. Let $\mathbf{M} = (A, B, Q, q_0, \lambda, \mu)$ be a finite-state sequential circuit. We say that $\mathbf{M}$ is a *simple non-Abelian Boolean circuit* if $A = B = \{0,1\}$, $\mu(Q) = \{0,1\}$, and $\mathbf{M}^S$ as a subsemigroup of $\mathbf{F}(Q)$ is a transitive simple non-Abelian group which is generated by two elements.

From the theory of permutation groups [4], all simple non-Abelian Boolean circuits can be constructed in the following way: let $\mathbf{G}$ be a finite simple non-Abelian group generated by the elements g_0 and g_1. Let $\mathbf{H} \leq \mathbf{G}$ be a subgroup. Let us consider the right cosets of $\mathbf{H}$ in $\mathbf{G}$: let $R = \{\mathbf{H}g : g \in \mathbf{G}\}$. Let $\mu\colon R \to \{0,1\}$ with $\mu(R) = \{0,1\}$ be arbitrary. Then $\mathbf{M} = (\{0,1\}, \{0,1\}, R, \mathbf{H}, \lambda, \mu)$ is a simple non-Abelian Boolean circuit where $\lambda(\mathbf{H}g, k) = \mathbf{H}gg_k$ for $k = 0, 1$.

Remark 138. Krohn, Maurer and Rhodes in [22] consider only those circuits for which $\mathbf{G}$ acts on Q primitively, in order to ensure that the size of the circuit (i.e. the number of states) is small.

We are especially interested in the following circuit corresponding to the group $\mathbf{A}_m$ for $m \geq 5$: let $\mathbf{H} = \mathbf{A}_{m-1} \leq \mathbf{A}_m = \{\pi \in \mathbf{A}_m : \pi(m) = m\}$ is

the stabilizer subgroup of the element m. For $2 \nmid m$, let $g_0 = (1\,2\,3)$ and let $g_1 = (3\,4\,\ldots\,m)$. For $2 \mid m$, let $g_0 = (1\,2\,3)$ and let $g_1 = (1\,2)\,(3\,4\,\ldots\,m)$. Then g_0 and g_1 generates $\mathbf{A}_m$ (see e.g. [4]). Finally let $\mu\colon R \to \{0,1\}$ be arbitrary such that $\mu(\mathbf{H}g_0) = 0$ and $\mu(\mathbf{H}g_1) = 1$ (such μ exists, since $g_0 \in \mathbf{H}$ and $g_1 \notin \mathbf{H}$).

Now we define how Boolean functions correspond to special polynomials over $\mathbf{G}$:

Definition 139. Let $\mathbf{G}$ be a finite simple non-Abelian group, where the elements g_0 and g_1 generate $\mathbf{G}$. Let $\mathbf{M} = (\{0,1\}, \{0,1\}, R, \mathbf{H}, \lambda, \mu)$ be a simple non-Abelian Boolean circuit. Let p be an n-ary polynomial over $\mathbf{G}$ which contains no inverses and every constant occurring in p is either g_0 or g_1. Then $B(\mathbf{M}, p) : \{0,1\}^n \to \{0,1\}$ is the Boolean function of n variables such that

$$B(\mathbf{M},p)(y_1,\ldots,y_n) = \mathbf{M}_{\mathbf{H}}(p(g_{y_1},\ldots,g_{y_n})) = \mu\big(\lambda'(p(g_{y_1},\ldots,g_{y_n}))(\mathbf{H})\big).$$

The value attained by the function $B(\mathbf{M}, p)$ at the input n-tuple $(y_1, \ldots, y_k)$ is nothing else than the output what the machine $\mathbf{M}$ attains for the word $p(x_1, \ldots, x_n)$, where $x_j = q_0$ if $y_j = 0$ and $x_j = q_1$ if $y_j = 1$.

Krohn, Rhodes and Maurer in [22] proved that for every finite simple non-Abelian circuit $\mathbf{M}$ and for any n-ary Boolean function f there exists a polynomial p over $\mathbf{G}$ such that $f = B(\mathbf{M}, p)$. They, however, did not investigate how long such a p must be. In the main theorem of the Section we use the results of Chapter 3 for giving an upper bound on $\|p\|$.

Theorem 140. *Let $\mathbf{G}$ be a finite simple non-Abelian group, where the elements g_0 and g_1 generate $\mathbf{G}$. Let K be the number of conjugacy classes of $\mathbf{G}$ and let $N = |\mathbf{G}|$. Let $\mathbf{M} = (\{0,1\}, \{0,1\}, R, \mathbf{H}, \lambda, \mu)$ be a simple non-Abelian Boolean circuit such that $\mu(R) = \{0,1\}$. Let $f\colon \{0,1\}^n \to \{0,1\}$ be an arbitrary function with e-many non-zero values. Then there exists a polynomial p over $\mathbf{G}$ such that p does not contain inverses, every constant in p is either g_0 or g_1, $f = B(\mathbf{M}, p)$, and*

$$\|p\| \leq 1\,605\,632 \cdot (N-1) \cdot (K-1)^2 \cdot n^8 \cdot e + (N-1).$$

If $\mathbf{G} = \mathbf{A}_m$ ($m \geq 5$), $\mathbf{H} = \mathbf{A}_{m-1}$, $g_0 = (1\,2\,3)$, and $g_1 = (3\,\ldots\,m)$ (if $2 \nmid m$) or $g_1 = (1\,2)\,(3\,\ldots\,m)$ (if $2 \mid m$) then we can choose p, such that

$$\|p\| \leq 128 \cdot \lfloor m/2 \rfloor \cdot n^2 \cdot e + (N-1).$$

Proof. Let $u_0, u_1 \in \mathbf{G}$ be elements such that $\mu(\mathbf{H}u_0) = 0$ and $\mu(\mathbf{H}u_1) = 1$. Since $\mathbf{G}$ is functionally complete, we can find an n-ary polynomial p' over

$\mathbf{G}$ such that $p'(g_{j_1}, \ldots, g_{j_n}) = u_j$, whenever $f(j_1, \ldots, j_n) = j$. Moreover by Remark 77 choosing $S_1 = \cdots = S_n = \{1, g_0, g_1\}$ and $S = \{g_0, g_1\}$ we have

$$v(p') \leq 3\,136 \cdot (K-1)^2 \cdot 2^8 \cdot n^8 \cdot e = 802\,816 \cdot (K-1)^2 \cdot n^8 \cdot e.$$

Now p' might contain inverses and constants apart from g_0 and g_1. For every occurrence of x_j^{-1} (for every $1 \leq j \leq n$) we replace x_j^{-1} by x_j^{N-1}. Moreover for every constant c appearing in p' we replace c by a product t_c of g_0 and g_1 such that $t_c = c$. Thus we obtain a polynomial p such that p does not contain inverses, every constant in p is either g_0 or g_1, and $f = B(\mathbf{M}, p)$. All that remains is to give an upper bound on $\|p\|$.

Let us define the following sequence of sets: let T_j contain every element of G which can be obtained by multiplying j-many elements from the set $\{g_0, g_1\}$. Now $T_1 \subseteq T_2 \subseteq \cdots \subseteq T_N$ and if $T_{j-1} \subsetneqq T_j$, then $1 + |T_{j-1}| \leq |T_j|$. Since g_0 and g_1 generate $\mathbf{G}$ and $|T_1| = 2$ we have $T_{N-1} = G$. Applying Proposition 74 we have

$$\|p\| \leq (N-1) \cdot \|p'\| \leq (N-1) \cdot (2 \cdot v(p') + 1),$$

from which we obtain the desired bound.

Now let us suppose that $\mathbf{G} = \mathbf{A}_m$, $\mathbf{H} = \mathbf{A}_{m-1}$ and $g_0 = (1\,2\,3)$. Let us choose $u_0 = g_0$ and $u_1 = g_1$. Similarly as before we can choose p' such that $p'(g_{j_1}, \ldots, g_{j_n}) = u_j = g_j$, whenever $f(j_1, \ldots, j_n) = j$. By Remark 77 choosing $S_1 = \cdots = S_n = \{1, g_0, g_1\}$ and $S = \{g_0, g_1\}$ we have

$$v(p') \leq K_{b,\{g_0,g_1\}} \cdot K_{\{g_0,g_1\},b} \cdot v\left(f_b^{(2)}\right)^3 \cdot n^{\log v\left(f_b^{(2)}\right)} \cdot e,$$

for some $b \in \mathbf{A}_m$. Let us choose $b = g_0$, then by Proposition 92 we have $v\left(f_b^{(2)}\right) = 4$. Clearly $K_{\{g_0,g_1\},g_0} = 1$, and by Proposition 97 we have $K_{g_0,\{g_0,g_1\}} \leq \lfloor m/2 \rfloor$. Therefore

$$v(p') \leq 64 \cdot \lfloor m/2 \rfloor \cdot n^2 \cdot e.$$

Similarly as above, we can obtain a polynomial p such that p does not contain inverses, every constant in p is either g_0 or g_1, $f = B(\mathbf{M}, p)$ and

$$\|p\| \leq (N-1) \cdot \|p'\| \leq (N-1) \cdot (2 \cdot v(p') + 1),$$

which gives us the desired bound. □

If applying an element of $\mathbf{G}$ on the machine $\mathbf{M}$ takes one time-step, then $\|p\|$ is the time required for calculating the function f with the machine $\mathbf{M}$. This is an alternative way of representing Boolean functions than what we

introduced in Section 4.1. Our upper bound on $\|p\|$, however, does not seem to be any better than that in Corollary 119. This might suggest that this representation is not better than the circuit-representation. There are examples, however, when the circuit-representation is less efficient, e.g. Krohn, Maurer and Rhodes in [22] represent the function $f\colon \{0,1\}^3 \to \{0,1\}$, $f(x_1,x_2,x_3) = x_1+x_2+x_3$ by a polynomial p with $\|p\| = 4$ over $\mathbf{A}_5$. On the other hand, $s_{\mathbf{B}}(f) = 6$ (see e.g. Theorem 3.1 on page 125 in [40]). Therefore there are situations when the method presented in this Section can be more efficient than the circuit representation.

4.6 Problems

Several gaps in our knowledge remain to be filled. One of the most interesting is whether the method for simulating the ring $\mathbf{Z}_p$ with the alternating group $\mathbf{A}_m$ (for $m \geq p+2$) can be extended to other rings.

Problem 3. Find a way of efficiently simulating an arbitrary ring $\mathbf{R}$ by a $\mathbf{G}$-circuit.

In Section 4.5 we investigated the efficiency of finite-state sequential circuits. We observed that in general it seems to be less efficient to realize a function by finite-state sequential circuits rather than by the two-element Boolean algebra $\mathbf{B}$. On the other hand, we showed a function which can be realized more efficiently using the finite-state sequential machines. More of such examples would be naturally welcome.

Problem 4. For a finite simple non-Abelian group $\mathbf{G}$ characterize the n-ary functions $f\colon \{0,1\}^n \to \{0,1\}$ which can be represented more efficiently by $\mathbf{G}$-circuits or by finite-state sequential circuits over $\mathbf{G}$ than by the two-element Boolean algebra $\mathbf{B}$.

Chapter 5

Complexity and functionally complete algebras

Up to this point we were examining the situation when a function or partial function was given over a functionally complete algebra and we had to find some polynomials which realize this function. While in Chapter 3 we gave upper and lower bounds on the length of a shortest realizing polynomial, in Chapter 4 we were considering computational models and studied fastest ways to compute the given function.

There are situations when one has to deal with polynomials directly. In such a situation it is important to know what function does the polynomials realize. From now on we consider two main versions of this problem. The first problem is called the *polynomial equivalence problem*, when one has to decide, whether or not two polynomials realize the same function. If both polynomials are terms (i.e. polynomials without any constants from the algebra) then we call it the *equivalence problem* or *identity checking problem*. The other problem is the *polynomial equation satisfiability problem* or *polynomial equation solvability problem*, when one has to decide whether the two polynomials attain the same value for at least one substitution. Among classical algebras (like groups or rings) this problem is trivial if neither of the polynomials have constants (and the answer is always 'yes', not depending on the two terms). Therefore we leave the word 'polynomial' out from the name of this problem. Compared to function realization problems, the equivalence and the equation solvability problems make sense not only over functionally complete algebras, but over any finite algebra.

These problems are all decidable questions for a finite algebra, the interesting question to ask is how hard is or how long it takes to decide them. Therefore we check the computational complexity of these questions.

Let us start with a notation. To every term or polynomial expression

$t(x_1, \ldots, x_n)$ and each algebra $\mathbf{A}$ we denote the naturally associated function by $t^{\mathbf{A}} \colon A^n \to A$. We recall that an algebra $\mathbf{A}$ *satisfies* an equation $s(\vec{x}) \approx t(\vec{x})$ for $\vec{x} = (x_1, \ldots, x_n)$, if the corresponding functions $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are the same function. We denote it by $\mathbf{A} \models s \approx t$.

Definition 141. *Equivalence problem* and *polynomial equivalence problem.*

Given: A finite algebra $\mathbf{A}$.

Instance: Two term expressions (for the equivalence problem), or two polynomial expressions (for the polynomial equivalence problem). Let the two expressions be s and t.

Question: Do the two input expressions realize the same function over $\mathbf{A}$, i.e. does $\mathbf{A} \models s \approx t$ hold?

Definition 142. *Equation solvability problem.*

Given: A finite algebra $\mathbf{A}$.

Instance: Two polynomial expressions p, q.

Question: Do the two input polynomials attain the same value for at least one substitution over $\mathbf{A}$, i.e. does the equation $p = q$ have a solution over $\mathbf{A}$?

We investigate these problems from Chapter 5 to Chapter 8. We start with the case when the algebra is functionally complete.

In Theorem 6 on page 752 of [29] Tobias Nipkow asserted the following:

Theorem 143. *The equation solvability problem for a nontrivial functionally complete algebra* $\mathbf{A}$ *is NP-complete.*

In the 'proof' he claims to give a polynomial reduction from deciding whether an equation over $\mathbf{Z}_2 = (\{ 0, 1 \}, +, \cdot)$ has a solution (a problem which is well-known to be NP-complete, see e.g. [7]) to the problem of whether an equation over $\mathbf{A}$ has a solution. Following the original proof from [29] shows that Nipkow's construction actually yields a reduction to the problem of whether a *system* of equations over $\mathbf{A}$ has a solution, which proves a weaker theorem:

Theorem 144. *The system of equations solvability problem for a nontrivial functionally complete algebra* $\mathbf{A}$ *is NP-complete.*

The definition of this problem is the following:

Definition 145. *System of equations solvability problem.*

Given: A finite algebra $\mathbf{A}$.

Instance: A natural number n and two system of polynomials $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ over $\mathbf{A}$.

Question: Does the system of equations $p_1 = q_1, \ldots, p_n = q_n$ have a solution over $\mathbf{A}$?

In Section 5.1 we first give the original proof from [29] (with slight modifications) yielding Theorem 144. Then in Section 5.2 we prove the theorem that Nipkow intended to prove. Finally in Section 5.3 we prove the following corollary of the method:

Theorem 146. *The polynomial equivalence problem for a nontrivial functionally complete algebra* $\mathbf{A}$ *is coNP-complete.*

5.1 The complexity of system of equations solvability problem

We give the proof of Theorem 144 in this Section.

Let $\mathbf{A}$ be a nontrivial functionally complete algebra ($|\mathbf{A}| \geq 2$). The problem is in NP, since we only need to substitute a possible solution.

It is well-known (see, e.g. [7] p. 251, problem AN9) that deciding whether an equation over $\mathbf{Z}_2 = (\{0, 1\}, +, \cdot)$ has a solution is NP-complete (it is almost the same as the SAT problem). Following the proof in [29] we give a polynomial reduction from the problem of determining whether an equation over $\mathbf{Z}_2$ has a solution to the problem of whether a system of equations over $\mathbf{A}$ has a solution.

Let $f(\underline{x}) = g(\underline{x})$ be an equation over $\mathbf{Z}_2$, where f and g are polynomial expressions and $\underline{x}$ is an n-tuple of free variables. We create a system of equations over $\mathbf{A}$ in polynomial time such that the system has a solution over $\mathbf{A}$ if and only if $f = g$ has a solution over $\mathbf{Z}_2$. The size of the system will be polynomial in $\|f\| + \|g\|$.

Let us denote two arbitrary distinct elements of $\mathbf{A}$ with $0_\mathbf{A}$ and $1_\mathbf{A}$. Since $\mathbf{A}$ is functionally complete, there exist two 2-variable polynomial expressions (let us denote them with $+_\mathbf{A}$ and $\cdot_\mathbf{A}$) such that $0_\mathbf{A}$ and $1_\mathbf{A}$ behave under the operations $+_\mathbf{A}$ and $\cdot_\mathbf{A}$ as 0 and 1 behave under the operations $+$ and $\cdot$, namely:

$+_\mathbf{A}(0_\mathbf{A}, 0_\mathbf{A}) = +_\mathbf{A}(1_\mathbf{A}, 1_\mathbf{A}) = 0_\mathbf{A}$, $+_\mathbf{A}(0_\mathbf{A}, 1_\mathbf{A}) = +_\mathbf{A}(1_\mathbf{A}, 0_\mathbf{A}) = 1_\mathbf{A}$,

$\cdot_{\mathbf{A}}(0_{\mathbf{A}}, 0_{\mathbf{A}}) = \cdot_{\mathbf{A}}(0_{\mathbf{A}}, 1_{\mathbf{A}}) = \cdot_{\mathbf{A}}(1_{\mathbf{A}}, 0_{\mathbf{A}}) = 0_{\mathbf{A}}$, and $\cdot_{\mathbf{A}}(1_{\mathbf{A}}, 1_{\mathbf{A}}) = 1_{\mathbf{A}}$.

There exist many possible functions for $+_{\mathbf{A}}$ and for $\cdot_{\mathbf{A}}$, and each can be expressed as a polynomial expression. We choose $+_{\mathbf{A}}$ and $\cdot_{\mathbf{A}}$ arbitrarily (with respect to these properties) and fix them for the proof.

There exists a 1-variable expression $\chi_{1_{\mathbf{A}}}$ such that $\chi_{1_{\mathbf{A}}}(1_{\mathbf{A}}) = 1_{\mathbf{A}}$ and $\chi_{1_{\mathbf{A}}}(a) = 0_{\mathbf{A}}$ for every $a \neq 1_{\mathbf{A}}$. Now using $+_{\mathbf{A}}$ and $\cdot_{\mathbf{A}}$ instead of $+$ and $\cdot$ and using $\chi_{1_{\mathbf{A}}}(x_i)$ instead of the variable x_i we can encode the equation $f = g$ over $\mathbf{Z}_2$ as an equation $f_{\mathbf{A}} = g_{\mathbf{A}}$ over $\mathbf{A}$ such that $f = g$ has a solution over $\mathbf{Z}_2$ if and only if $f_{\mathbf{A}} = g_{\mathbf{A}}$ has a solution over $\mathbf{A}$. We can observe though that if we want to express this equation using the basic operations of $\mathbf{A}$ then the length of the resulting equation might be exponential in the size of the original equation (e.g. if any variable occurs more than once in the polynomial expression for $+_{\mathbf{A}}$ or for $\cdot_{\mathbf{A}}$).[1] For this reason, the proof is not a polynomial reduction from deciding whether an equation over $\mathbf{Z}_2$ has a solution to deciding whether an equation over $\mathbf{A}$ has a solution. However, using an easy trick we can encode the original equation to a *system of equations* with polynomial size in $\|f\| + \|g\|$:

At first we have the equation $f(\underline{x}) = g(\underline{x})$ over $\mathbf{Z}_2$. In every step we will shorten this equation and add other equations to our system until the equation cannot be shortened any more. In each step we search reading from left to right in our modified equation for any occurrence of $x + y$ or of $x \cdot y$, where x and y are variables or constants (polynomial expressions with length 1). If we find an occurrence of $x + y$ with variables or constants x, y then for a new variable z we replace every occurrence of $x + y$ with z in the modified equation and add the equation $z = +_{\mathbf{A}}(x, y)$ to our system of equations. Similarly, if we find an occurrence of $x \cdot y$ with variables or constants x, y then for a new variable z we replace every occurrence of $x \cdot y$ with z in the modified equation and add the equation $z = \cdot_{\mathbf{A}}(x, y)$ to our system of equations. Each step takes at most $\|f\| + \|g\|$ time and each step shortens the equation $f = g$, hence the algorithm stops in at most $(\|f\| + \|g\|)^2$ time. After the final step, in every equation of the system for every *original* variable x_i (i.e. which occurred in $f = g$) we replace x_i with $\chi_{1_{\mathbf{A}}}(x_i)$.

After this translation we have a system of equations over $\mathbf{A}$ such that the system has a solution over $\mathbf{A}$ if and only if the original equation $f = g$ had a solution over $\mathbf{Z}_2$. The size of the system is linear in the size of the equation $f = g$ over $\mathbf{Z}_2$, since there are at most $(\|f\| + \|g\|)$-many equations, and by Lemma 39 each equation has length at most $(\|+_{\mathbf{A}}\| + \|\cdot_{\mathbf{A}}\|) \cdot \|\chi_{1_{\mathbf{A}}}\|$, which

[1] An easy example for such an exponential blowup is if for a group one wants to express the commutator expression $[[[[x_1, x_2], x_3] \dots], x_n]$ using only the inverse operation and the multiplication of the group.

does not depend on the equation but on the algebra $\mathbf{A}$. The time of the translation of $f = g$ over $\mathbf{Z}_2$ to a system of equations over $\mathbf{A}$ is polynomial as well, which finishes the proof.

5.2 The complexity of the equation solvability problem

We give the proof of Theorem 143 in this Section.

Let $\mathbf{A}$ be a nontrivial functionally complete algebra ($|\mathbf{A}| \geq 2$). The problem is in NP, since we only need to substitute a possible solution.

It is well-known (see, e.g. [7]) that deciding whether a formula written in conjunctive normal form can be satisfied over the two-element Boolean algebra $\mathbf{B} = (\{0, 1\}, \neg, \vee, \wedge)$ is NP-complete (this is called the SAT problem). The formula is usually given by the clauses, which we take the conjunctions of, where each clause is a disjunction of arbitrary many literals, i.e. variables or negations of variables ([7] p. 259 problem LO1). The problem remains NP-complete, if every clause in the conjunctive normal form contains exactly 3 literals (this is called the 3SAT problem, [7] p. 259 problem LO2). We will give a polynomial reduction from the problem of determining whether a 3SAT formula can be satisfied over $\mathbf{B}$ to the problem of whether an equation over $\mathbf{A}$ has a solution.

Let $\varphi(\underline{x}) = \bigwedge_{i=1}^{n} p_i$ be a 3SAT formula over $\mathbf{B}$. We create an equation over $\mathbf{A}$ such that the equation has a solution over $\mathbf{A}$ if and only if φ can be satisfied over $\mathbf{B}$. The length of the equation will be polynomial in the size of the formula.

Let us denote two arbitrary distinct elements of $\mathbf{A}$ with $0_{\mathbf{A}}$ and $1_{\mathbf{A}}$. Since $\mathbf{A}$ is functionally complete, there exists a 2-variable polynomial expression $\wedge_{\mathbf{A}}$ such that $0_{\mathbf{A}}$ and $1_{\mathbf{A}}$ behave under the operation $\wedge_{\mathbf{A}}$ as 0 and 1 behave under the operation $\wedge$, namely $\wedge_{\mathbf{A}}(0_{\mathbf{A}}, 0_{\mathbf{A}}) = \wedge_{\mathbf{A}}(0_{\mathbf{A}}, 1_{\mathbf{A}}) = \wedge_{\mathbf{A}}(1_{\mathbf{A}}, 0_{\mathbf{A}}) = 0_{\mathbf{A}}$, and $\wedge_{\mathbf{A}}(1_{\mathbf{A}}, 1_{\mathbf{A}}) = 1_{\mathbf{A}}$. There exist many possible functions for $\wedge_{\mathbf{A}}$, and each can be expressed as a polynomial expression. We choose $\wedge_{\mathbf{A}}$ arbitrarily (with respect to these properties) and fix it for the proof. Similarly, for each of the eight possible 3-variable forms of disjunctive clause $q_j = q_j(x_1, x_2, x_3)$, $(j = 1, \ldots, 8)$ we can choose an arbitrary but fixed 3-variable expression $q_{j,\mathbf{A}}$ such that $0_{\mathbf{A}}$ and $1_{\mathbf{A}}$ behave under the function $q_{j,\mathbf{A}}$ as 0 and 1 behave under the clause q_j. Moreover there exists a 1-variable expression $\chi_{1_{\mathbf{A}}}$ such that $\chi_{1_{\mathbf{A}}}(1_{\mathbf{A}}) = 1_{\mathbf{A}}$ and $\chi_{1_{\mathbf{A}}}(a) = 0_{\mathbf{A}}$ for every $a \neq 1_{\mathbf{A}}$.

For every positive integer number k we will use a polynomial $\wedge^{(k)} = \wedge^{(k)}_{\mathbf{A}}(x_1, \ldots, x_k)$ over $\mathbf{A}$ in a way that it behaves on inputs from $\{0_{\mathbf{A}}, 1_{\mathbf{A}}\}$

the very same as $\bigwedge_{i=1}^{k} x_i$ behaves on the inputs $\{0,1\}$ over $\mathbf{B}$. Let us define $\wedge^{(k)}$ in the same way we defined the polynomials $p^{(n)}$ in Lemma 44: let $\wedge_{\mathbf{A}}^{(1)}(x_1) = x_1$ and $\wedge_{\mathbf{A}}^{(2)}(x_1, x_2) = \wedge_{\mathbf{A}}(x_1, x_2)$. For every integer $i \geq 2$ let

$$\begin{aligned} \wedge_{\mathbf{A}}^{(2i-1)}(x_1, \ldots, x_{2i-1}) &= \wedge_{\mathbf{A}}^{(2)}\left(\wedge_{\mathbf{A}}^{(i)}(x_1, \ldots, x_i), \wedge_{\mathbf{A}}^{(i-1)}(x_{i+1}, \ldots, x_{2i-1})\right), \\ \wedge_{\mathbf{A}}^{(2i)}(x_1, \ldots, x_{2i}) &= \wedge_{\mathbf{A}}^{(2)}\left(\wedge_{\mathbf{A}}^{(i)}(x_1, \ldots, x_i), \wedge_{\mathbf{A}}^{(i)}(x_{i+1}, \ldots, x_{2i})\right). \end{aligned}$$

It is clear that $\wedge_{\mathbf{A}}^{(k)}$, for every integer k, has the required property.

Now using the expression $q_{j,\mathbf{A}}$ instead of the clause q_j, using $\wedge_{\mathbf{A}}^{(n)}$ instead of $\wedge_{i=1}^{n}$ and using $\chi_{1_{\mathbf{A}}}(x_i)$ instead of the variable x_i we can encode the formula φ over $\mathbf{B}$ as an expression $\varphi_{\mathbf{A}}$ over $\mathbf{A}$ such that φ can be satisfied over $\mathbf{B}$ if and only if $\varphi_{\mathbf{A}} = 1_{\mathbf{A}}$ has a solution over $\mathbf{A}$. The only remaining part is to prove that $\|\varphi_{\mathbf{A}}\|$ is polynomial in $\|\varphi\|$.

Let $c = \|\chi_{1_{\mathbf{A}}}\|$, let $l = \|\wedge_{\mathbf{A}}\|$ and let $d = \max\{\|q_{j,\mathbf{A}}\| : j = 1, \ldots, 8\}$ the length of the longest clause expression. For every k we have $\left\|\wedge_{\mathbf{A}}^{(k)}\right\| \leq l^{\lceil \log k \rceil} \leq l \cdot k^{\log l}$, which is quite straightforward from Lemma 44 or from the fact $\left\|\wedge_{\mathbf{A}}^{(k)}\right\| \leq \left\|\wedge_{\mathbf{A}}^{(2)}\right\| \cdot \max\left\{\left\|\wedge_{\mathbf{A}}^{(\lceil k/2 \rceil)}\right\|, \left\|\wedge_{\mathbf{A}}^{(\lfloor k/2 \rfloor)}\right\|\right\}$.

Using Lemma 39 we can conclude that the length of the expressed 3SAT formula $\varphi_{\mathbf{A}}$ over $\mathbf{A}$ is not more than $c \cdot d \cdot l \cdot n^{\log l}$, which is polynomial in the length of the original 3SAT formula $\|\varphi\|$, since $n \leq \|\varphi\|$ and c, d, l depend only on $\mathbf{A}$. Thus, Theorem 143 is recovered.

5.3 The complexity of the polynomial equivalence problem

With a slight modification we can easily prove Theorem 146. Let $\mathbf{A}$ be a nontrivial functionally complete algebra ($|\mathbf{A}| \geq 2$). The problem is in coNP, since we only need to substitute a possible counterexample.

In the proof of Theorem 143, for every 3SAT formula φ we created an expression $\varphi_{\mathbf{A}}$ over $\mathbf{A}$ such that φ can be satisfied over $\mathbf{B}$ if and only if $\varphi_{\mathbf{A}} = 1_{\mathbf{A}}$ has a solution over $\mathbf{A}$. Moreover the length of $\varphi_{\mathbf{A}}$ was polynomial in the length of φ. Observe that the image of $\varphi_{\mathbf{A}}$ over $\mathbf{A}$ is a (not necessarily proper) subset of $\{0_{\mathbf{A}}, 1_{\mathbf{A}}\}$, hence $\varphi_{\mathbf{A}} = 1_{\mathbf{A}}$ has a solution over $\mathbf{A}$ if and only if $\varphi_{\mathbf{A}} \approx 0_{\mathbf{A}}$ is *not* an identity over $\mathbf{A}$. This is a polynomial reduction from the problem of 3SAT over $\mathbf{B}$ to the problem of determining whether an equation is an identity over $\mathbf{A}$.

Chapter 6

The complexity of the polynomial equivalence problem for meta-Abelian groups

Having investigated the polynomial equivalence and equation solvability problems for functionally complete algebras, we turn our attention to classical algebraic structures.

Early investigations into the equivalence problem for various finite algebraic structures were carried out by computer scientists, in particular at Syracuse University where the terminology *the term equivalence problem* was introduced. They considered finite commutative rings and finite lattices. In the early 1990's it was shown by Hunt and Stearns (see [16]) that the equivalence problem of a finite commutative ring either has polynomial time complexity or is coNP-complete. Later Burris and Lawrence proved in [2] that the same holds for rings in general.

Theorem 147. *Let* $\mathbf{R}$ *be a finite ring. The equivalence problem for* $\mathbf{R}$ *is in* P *if* $\mathbf{R}$ *is nilpotent, and it is coNP-complete otherwise.*

It is not hard to see that from the proof the same follows for the polynomial equivalence problem. Surprisingly enough there are no published results about the complexity of the equation solvability problem for finite rings.

The equivalence problem for finite groups has proved to be a far more challenging topic than that for finite rings. This problem for a group $\mathbf{G}$ is the problem of deciding which equations $s \approx t$ are satisfied by $\mathbf{G}$. We recall a notation from Chapter 5. To every term or polynomial expression $t(x_1, \ldots, x_n)$ and each group $\mathbf{G}$ we denote the naturally associated function by $t^{\mathbf{G}} \colon G^n \to G$. We recall that a group $\mathbf{G}$ *satisfies* an equation $s(\vec{x}) \approx t(\vec{x})$

for $\vec{x} = (x_1, \ldots, x_n)$, if the corresponding functions $s^{\mathbf{G}}$ and $t^{\mathbf{G}}$ are the same function. We denote it by $\mathbf{G} \models s \approx t$. We recall that $\mathbf{G} \models s \approx t$ if and only if $\mathbf{G} \models s \cdot t^{-1} \approx 1$. Therefore we view the equivalence problem for groups as the problem of deciding which equations $t \approx 1$ are satisfied by $\mathbf{G}$.

In 2004 Burris and Lawrence [3] proved that if $\mathbf{G}$ is nilpotent or $\mathbf{G} \simeq \mathbf{D}_n$, the dihedral group for odd n, then the polynomial equivalence problem for $\mathbf{G}$ is in P. The groups arising for the next step of the investigation are the meta-Abelian groups.

This Chapter investigates the case of meta-Abelian groups. We prove that for several kinds of semidirect products the polynomial equivalence problem is in P. Examples for such groups are the above-mentioned dihedral groups, the alternating group $\mathbf{A}_4$, or the wreath product of two cyclic group.

From Theorem 146 in Chapter 5 we already know that the polynomial equivalence problem is coNP-complete for finite simple non-Abelian groups. The result does not tell us anything about the complexity of the equivalence problem as it uses the constants of the group. In Chapter 7 we prove that not only for the simple non-Ableian groups but for every finite nonsolvable group the equivalence problem is coNP-complete.

Interest in the computational complexity of the equivalence problem of a finite algebraic structure has been steadily increasing since 2004. There are many results about the equivalence problem of finite monoids [21], [37], [38]. Their initial approach came from the complexity of recognizing formal languages. The first hardness result for semigroups was proved by Popov and Volkov [39], and several results were proved by Seif and Szabó in [34]. For commutative semigroups the topic was thoroughly investigated by Kisielewicz [19].

The complexity of the system of equation solvability problem is completely characterized for groups in [10] and [23]. For a finite Abelian group deciding whether a system of equations has a solution is in P, otherwise it is NP-complete.

The characterization of solving a single equation looks more complicated, though ([10]). Goldmann and Russell proved that for a finite group $\mathbf{G}$ deciding whether an equation has a solution is in P if $\mathbf{G}$ is nilpotent and NP-complete if $\mathbf{G}$ is non-solvable.

The result tells nothing about non-nilpotent solvable groups. Goldmann and Russell explicitly ask in [10] to decide the complexity of solving an equation over $\mathbf{S}_3$.

The equation solvability problem was first examined for monoids and semigroups. Klíma [20] has analyzed the question for semigroups of size at most 6. He proved for almost all of these semigroups that solving an equation

is in either in P or NP-complete. The only remaining case is the 6 element 'monoid' $\mathbf{S}_3$. He conjectures that the problem is in P.

In Section 6.2 we show the following: If $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$, where $\mathbf{A} \simeq \mathbf{Z}_p$ and $\mathbf{B} \simeq \mathbf{Z}_q$ for some primes p and q, then equation solvability problem is in P. Thus, with $\mathbf{Z}_3 \simeq \mathbf{A}$ and $\mathbf{Z}_2 \simeq \mathbf{B}$ we answer the questions of Goldmann, Russell and Klíma.

The results suggest that the complexity of equivalence problem for a finite algebra $\mathbf{A}$ is in P if and only if the equation solvability problem for $\mathbf{A}$ is in P. This is far from to be true. Seif and Szabó presented a 10 element semigroup (see [34]) for which the equivalence problem is in P and the equation solvability problem is NP-complete. Klíma proved an even stronger result in [20], where he showed a semigroup of size 24 for which the equation solvability problem is NP-complete but the polynomial equivalence problem is in P.

It may happen, though, that the complexity of the two problems coincide in case of groups. At this point we do not even know these complexities for the symmetric group $\mathbf{S}_4$.

6.1 Semidirect products

In this Section we prove for a class of non-nilpotent groups that the polynomial equivalence problem (and so the equivalence problem) can be solved in polynomial time. The following method will play a crucial role in our investigation.

Collecting procedure: Let $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ where $\mathbf{A}$ is Abelian and let $t = x_1 x_2 \dots x_k$ be a group polynomial over $\mathbf{G}$. Without loss of generality we assume that the x_i are constants or variables over $\mathbf{G}$. Every element of $\mathbf{G}$ can be uniquely written of the form ba where $a \in \mathbf{A}$ and $b \in \mathbf{B}$. So we write x_i of the form $b_i a_i$ where $a_i \in \mathbf{A}$ and $b_i \in \mathbf{B}$. Collecting the elements of $\mathbf{B}$ to the left we obtain

$$t = (b_1 b_2 \dots b_k) \cdot \left(a_1^{b_2 b_3 \dots b_k} a_2^{b_3 \dots b_k} \dots a_{k-1}^{b_k} a_k \right).$$

This term is an identity if and only if both

$$b_1 b_2 \dots b_k$$

and

$$\left(a_1^{b_2 b_3 \dots b_k} a_2^{b_3 \dots b_k} \dots a_{k-1}^{b_k} a_k \right) \tag{6.1}$$

are identities (i.e. both are identically 1 for all substitutions over $\mathbf{G}$). Let us examine the latter expression. Substitute $a_i = 1$ for all i, where x_i was

a variable, not constant. Then we have $t' = c_1^{w_1} c_2^{w_2} \dots c_m^{w_m}$, where all c_is are constants from $\mathbf{A}$ and w_i is a word over $\mathbf{B}$ (let us call t' the *constant part* of (6.1)). Let us fix j. Substituting $a_i = 1$ for $i \neq j$ (where a_i is not constant) we obtain an identity of the form $t'_j t'$ where $t'_j = a_j^{h_1} a_j^{h_2} \dots a_j^{h_l}$ and l is the number of the occurrences of x_j in t and h_i is a semigroup polynomial over $\mathbf{B}$ for every $1 \leq i \leq l$. Obviously, (6.1) is an identity if and only if t' and t'_j are identities for every $1 \leq j \leq k$. Hence we are looking for the complexity of checking whether or not $b_1 b_2 \dots b_k$, t' and t'_j are all identities.

Lemma 148. *Let $\mathbf{F}$ be a field of prime characteristic p and let $\mathbf{H}$ be a multiplicative subgroup of $\mathbf{F}^*$: $\mathbf{H} \leq \mathbf{F}^*$. For a polynomial $f(\bar{x}) \in \mathbf{F}[x_1, x_2, \dots, x_k]$ it can be checked in polynomial time whether or not it vanishes on $\mathbf{H}$.*

Proof. Let a be a generator of $\mathbf{F}^*$ and let $\mathbf{H} = \langle a^t \rangle$. Putting $z_j = x_j^t$ we have $f(\bar{x})$ is identically 0 over $\mathbf{H}$ if and only if $f(\bar{z})$ is identically 0 over $\mathbf{F}^*$. A polynomial $g \in \mathbf{F}[x_1, \dots, x_k]$ admits this latter property if and only if $g = \sum (x_i^{q-1} - 1) g_i(\bar{x})$ for some $g_i \in \mathbf{F}[x_1, \dots, x_k]$, where $|\mathbf{F}| = q$. This condition can be checked in linear time since we only need to divide g by $x_i^{q-1} - 1$ (i.e. substitute $x_i^{q-1} = 1$) for all $i \in \{1, \dots, k\}$ and the remaining expression has to be 0. □

Theorem 149. *If $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ where $\mathbf{A} \simeq \mathbf{Z}_p$ for some prime p, and the polynomial equivalence problem for $\mathbf{B}$ is in P then the polynomial equivalence problem for $\mathbf{G}$ is in P, too.*

Proof. The subgroup $\mathbf{B}$ acts on $\mathbf{A}$. Now, Aut $\mathbf{A} \simeq \mathbf{C}_{p-1}$, the cyclic group of order $p-1$ and consists of the maps $a \to a^l$ for every $a \in \mathbf{A}$ for some $1 \leq l \leq p-1$. Thus there is a homomorphism $\phi : \mathbf{B} \to \mathbf{C}_{p-1}$ such that $a^b = a^{\phi(b)}$ for every $a \in \mathbf{A}$. Now, using the collecting procedure it is enough to check whether or not $b_1 b_2 \dots b_k$, $a_j^{h_1} a_j^{h_2} \dots a_j^{h_l}$ and $c_1^{w_1} c_2^{w_2} \dots c_m^{w_m}$ are identities. The first condition can be checked in polynomial time by the assumption. For the second one we rewrite the expression $a_j^{h_1} a_j^{h_2} \dots a_j^{h_l} = a_j^{\phi(h_1)} a_j^{\phi(h_2)} \dots a_j^{\phi(h_l)} = a_j^{w_1 + w_2 + \dots + w_l}$. Here w_j denotes the image of h_j at ϕ. Substituting $\phi(b_j) = y_j$ we have w_j as a product of some of $y_1, \dots y_k$ over $\mathbf{Z}_p$, shortly a monomial, and $f = w_1 + w_2 + \dots + w_l$ is a k-variable polynomial over $\phi(\mathbf{B})$ where both the addition and the multiplication is understood in $\mathbf{Z}_p$. The expression $a_j^{w_1 + w_2 + \dots + w_l}$ is an identity if and only if f attains 0 every time when we substitute elements of $\phi(\mathbf{B})$ for the variables. And this can be checked in polynomial time by Lemma 148. Finally, $c_1^{w_1} c_2^{w_2} \dots c_m^{w_m}$ can be written in the form $c^{w'_1} c^{w'_2} \dots c^{w'_m}$, where c is the generator, of $\mathbf{A}$. Using the same idea, this is an identity if and only if $w'_1 + \dots + w'_m$ attains 0 every time when we substitute elements of $\phi(\mathbf{B})$ for the variables. And this can be checked in polynomial time by Lemma 148, again. □

Corollary 150. *If* $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$, *where the polynomial equivalence problem for* $\mathbf{B}$ *is in P, and* $\mathbf{A} \simeq \mathbf{Z_m}$ *where* m *is squarefree, then the polynomial equivalence problem for* $\mathbf{G}$ *is in P, too.*

Proof. Now, $\mathbf{A} \simeq \oplus_{p|m}\mathbf{Z}_p$ and all summands are $\mathbf{B}$ invariant. Every constant can be uniquely decomposed into a product of elements from $\mathbf{Z}_p$ for $p|m$. For a polynomial p let $t_{(p)}$ denote the polynomial when we replace each constant by its p part. Obviously, a polynomial is an identity over $\mathbf{G}$ if and only if $t_{(p)}$ is an identity over $\mathbf{Z}_p \rtimes \mathbf{B}$ for every prime p dividing m. This can be checked in polynomial time by Theorem 149. □

Unfortunately the same idea does not work for a noncyclic normal subgroup, $\mathbf{A}$. The collecting procedure can be used in a few other cases, though.

Theorem 151. *Let* $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ *such that the following hold:*

(a) $\mathbf{A}$ *is Abelian and the exponent of* $\mathbf{A}$ *is squarefree;*

(b) the polynomial equivalence problem for $\mathbf{B}$ *is in P;*

(c) for ever prime p *dividing the size of* $\mathbf{A}$ *and* $\mathbf{P} \in Syl_p(\mathbf{A})$ *the group* $\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})$ *is Abelian and* $p \nmid |\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})|$, *where* $C_{\mathbf{B}}(\mathbf{P})$ *denotes the centralizer of* $\mathbf{P}$ *in* $\mathbf{B}$.

Then the polynomial equivalence problem for $\mathbf{G}$ *is in P.*

Proof. After the collection procedure we see that it is enough to check identities over $\mathbf{B}$ and identities of the form (6.1)

$$a^{x_1^{k_{11}}x_2^{k_{12}}\ldots x_n^{k_{1n}}}a^{x_1^{k_{21}}x_2^{k_{22}}\ldots x_n^{k_{2n}}}\ldots a^{x_1^{k_{l1}}x_2^{k_{l2}}\ldots x_n^{k_{ln}}}, \tag{6.2}$$

and $c_1^{w_1}c_2^{w_2}\ldots c_m^{w_m}$ for the constants. The Sylow subgroups of $\mathbf{A}$ are $\mathbf{B}$ invariant, hence it is enough to check the identity for the Sylows of $\mathbf{A}$. Thus we may assume that $\mathbf{A}$ is an elementary Abelian p-group. Let $\mathbf{A} \simeq \mathbf{Z}_p^m$ and let $\varphi\colon \mathbf{B} \to \text{Aut } \mathbf{Z}_p^m \simeq \mathbf{GL}_m(\mathbf{Z}_p)$ be the action of $\mathbf{B}$ on $\mathbf{A}$, $\varphi(\mathbf{B}) = \mathbf{H}$. With these notations we need to check identity (6.1) for $\mathbf{G} \simeq \mathbf{Z}_p^m \rtimes \mathbf{H}$, where $\mathbf{H}$ is an Abelian matrix group acting faithfully on $\mathbf{Z}_p^m$ (note that $\mathbf{H} \simeq \mathbf{B}/C_{\mathbf{B}}(\mathbf{Z}_p^m)$). Let $\mathbf{R}$ denote the subring of the ring of m by m matrices generated by $\mathbf{H}$. Now (6.2) can be rewritten as:

$$a^{x_1^{k_{11}}x_2^{k_{12}}\ldots x_n^{k_{1n}}+x_1^{k_{21}}x_2^{k_{22}}\ldots x_n^{k_{2n}}+\cdots+x_1^{k_{l1}}x_2^{k_{l2}}\ldots x_n^{k_{ln}}}$$

and it is enough to check whether or not the exponent

$$x_1^{k_{11}}x_2^{k_{12}}\ldots x_n^{k_{1n}} + x_1^{k_{21}}x_2^{k_{22}}\ldots x_n^{k_{2n}} + \cdots + x_1^{k_{l1}}x_2^{k_{l2}}\ldots x_n^{k_{ln}} \tag{6.3}$$

is identically 0 in $\mathbf{R}$ when substituting the elements of $\mathbf{H}$. The ring $\mathbf{R}$ acts semisimply on $\mathbf{Z}_p^m$, because $p \nmid |\mathbf{H}|$. By the Wedderburn–Artin Theorem [17] $\mathbf{R}$ is a direct sum of matrix-rings. As $\mathbf{H}$ is commutative, $\mathbf{R}$ is commutative, as well, hence $\mathbf{R}$ is a direct sum of fields: $\mathbf{R} = \oplus_{i=1}^{s} \mathbf{F}_{q_i}$. Thus $\mathbf{H} \leq \mathbf{R}^* \simeq \oplus_{i=1}^{s} \mathbf{F}_{q_i}^*$. Let $\mathbf{H}_i$ denote the projection of $\mathbf{H}$ to its i-th coordinate. Expression (6.3) is identically 0 over $\mathbf{R}$ if and only if it is 0 at every substitution from $\mathbf{H}_i$ for every $i \leq s$. By Lemma 148 this can be checked in polynomial time, and so the polynomial equivalence problem for $\mathbf{G}$ is in P.

Finally, consider the identity $c_1^{w_1} c_2^{w_2} \dots c_l^{w_m} \approx 1$. Here we can write every c_j as a linear combination of some fixed basis, $\{v_i\}$, of $\mathbf{A}$. Let $c_j = \prod v_i^{\lambda_{ji}}$. Thus, it is enough to check, whether $v_i^{\lambda_{1i} w_1} v_i^{\lambda_{2i} w_2} \dots v_i^{\lambda_{li} w_l} \approx 1$ is an identity for all $1 \leq i \leq s$. The exponent has to be identically 0 over $\mathbf{H_i}$, and this can be checked in polynomial time by Lemma 148. □

Corollary 152. *Let $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are Abelian groups, such that the exponent of A is squarefree and $(|\mathbf{A}|, |\mathbf{B}|) = 1$ then the polynomial equivalence problem for $\mathbf{G}$ is in P.*

Proof. The conditions of Theorem 151 trivially hold. □

Now, we investigate the case when neither the size nor the exponent of the normal subgroup is squarefree. The modification of the Lemma 148 remains valid for cyclic groups.

Lemma 153. *Let $f(x_1, \dots, x_k) = w_1 + \dots + w_l$ be a sum of monomials in k variables over $\mathbf{Z}_{p^\alpha}$ $(p > 2)$ and let $\mathbf{H}$ be the $p-1$ element subgroup of $\mathbf{Z}_{p^\alpha}^*$. Then, for any $\mathbf{M} \leq \mathbf{H}$ it can be checked in polynomial time whether or not f vanishes on $\mathbf{M}$.*

Proof. Let a be a generator of $\mathbf{H}$ and let $\mathbf{M} = \langle a^t \rangle$. Putting $z_j = x_j^t$ we have $f(\bar{x})$ is identically 0 over $\mathbf{M}$ if and only if $f(\bar{z})$ is identically 0 over $\mathbf{H}$. We claim that a polynomial $f \in \mathbf{Z}_{p^n}[x_1, \dots, x_k]$ admits this latter property if and only if $f = \sum (x_i^{p-1} - 1) g_i(\bar{x})$ for some $g_i \in \mathbf{Z}_{p^n}[x_1, \dots, x_k]$. This condition can be checked in linear time. Since the exponent of $\mathbf{H}$ is $p-1$, if f is of the required form, it vanishes over $\mathbf{H}$. On the other hand, as the elements of $\mathbf{H}$ are pairwise incongruent mod p (not only mod p^α), the polynomial has to vanish over $\mathbf{Z}_p^*$, as well. By Lemma 148 this happens if and only if $f = \sum (x_i^{p-1} - 1) g_{i1}(\bar{x}) \mod p$ and so $f = \sum (x_i^{p-1} - 1) g_{i1}(\bar{x}) + p f_1 \mod p^\alpha$. Hence f_1 is vanishing $\mod p^{\alpha-1}$. By the previous arguments $f_1 = \sum (x_i^{p-1} - 1) g_{i2}(\bar{x}) \mod p$. Continuing in the same fashion we obtain that $f = \sum (x_i^{p-1} - 1) g_i(\bar{x})$. □

The following theorem is a generalization of Theorem 149:

Theorem 154. *Let* $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ *such that the following hold:*

(a) $\mathbf{A}$ *is cyclic;*

(b) the polynomial equivalence problem for $\mathbf{B}$ *is in P;*

(c) for ever prime p *dividing the size of* $\mathbf{A}$ *and* $\mathbf{P} \in Syl_p(\mathbf{A})$ *we have* $p \nmid |\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})|$.

Then the polynomial equivalence problem for $\mathbf{G}$ *is in P.*

Proof. Going along the lines of Theorem 151, we may assume that $\mathbf{A} \simeq \mathbf{Z}_{p^m}$. Moreover, after the collection procedure, it is enough to check identities over $\mathbf{B}$ and identities of the form $f = w_1 + w_2 + \cdots + w_l = 0$ over $\mathbf{B}/C_{\mathbf{B}}(\mathbf{P})$ (Note that this works for the constant part, as well, since we can write every constant as a power of the generator of $\mathbf{A}$). As $\mathbf{B}/C_{\mathbf{B}}(\mathbf{P}) \leq$ Aut $\mathbf{Z}_{p^\alpha}$, condition (c) implies that $\mathbf{B}/C_{\mathbf{B}}(\mathbf{P}) \leq \mathbf{H}$, where $\mathbf{H}$ denotes the $p-1$ element subgroup of Aut $\mathbf{Z}_{p^\alpha}$. If $p = 2$ then $\mathbf{H} = 1$, if $p > 2$, then identities can be checked in polynomial time over $\mathbf{B}$ and $\mathbf{H}$, by condition (b), and by Lemma 153, respectively. □

6.2 Equation solvability

A modification of the collecting procedure and Lemma 148 will also help us to find out the complexity of the equation solvability problem for some metacyclic groups, including $\mathbf{S}_3$.

Theorem 155. *For any group* $\mathbf{G}$ *of order* pq *where* p *and* q *are primes the equation solvability problem for* $\mathbf{G}$ *is in P.*

Proof. Consider the case when $\mathbf{G} \simeq \mathbf{A} \rtimes \mathbf{B}$ where $\mathbf{A} \simeq \mathbf{Z}_p$ and $\mathbf{B} \simeq \mathbf{Z}_q$. We may assume that $\mathbf{G}$ is not abelian, and so $p \neq q$.

Let $\{t, s\}$ be an instance of the equation solvability problem for $\mathbf{G}$. We would like to know whether or not $t = s$ has a solution. Multiplying by s^{-1} and writing t for ts^{-1}, we have to solve $t = 1$. After the collecting procedure we obtain the following equation:

$$t(g_1 \dots g_k) = (b_1 b_2 \dots b_k) \cdot \left(a_1^{b_2 b_3 \dots b_k} a_2^{b_3 \dots b_k} \dots a_{k-1}^{b_k} a_k \right) = 1.$$

As p and q are coprime, both

$$b_1 b_2 \dots b_k = 1$$

and

$$a_1^{b_2 b_3 \dots b_k} a_2^{b_3 \dots b_k} \dots a_{k-1}^{b_k} a_k = 1.$$

must hold. Since $\mathbf{B}$ is cyclic, we can solve $b_1 \dots b_k = 1$ as a congruence mod q, and we can express one of the variables (say, b_1) using the other variables and constants: $b_1 = c \prod b_i^{k_i d}$, this is what a solution looks like mod q. Substituting this expression for b_1 in $t'_1 t'_2 \dots t'_k t' = 1$, we only need to check the complexity of the solvability of this latter equation under the constraint for b_1. By a similar argument as in the proof of Theorem 149 we arrive at the solvability of

$$a^{x_1^{k_{11}} x_2^{k_{12}} \dots x_n^{k_{1n}} + x_1^{k_{21}} x_2^{k_{22}} \dots x_n^{k_{2n}} + \dots + x_1^{k_{l1}} x_2^{k_{l2}} \dots x_n^{k_{ln}}} = 1,$$

where a is a generator of $\mathbf{A}$. Now, it is enough to check whether or not the exponent attains 0, that is whether or not

$$x_1^{k_{11}} x_2^{k_{12}} \dots x_n^{k_{1n}} + x_1^{k_{21}} x_2^{k_{22}} \dots x_n^{k_{2n}} + \dots + x_1^{k_{l1}} x_2^{k_{l2}} \dots x_n^{k_{ln}} = 0$$

has a solution over $\mathbf{Z}_p$. As p is a prime, this equation has no solution if and only if

$$(x_1^{k_{11}} x_2^{k_{12}} \dots x_n^{k_{1n}} + x_1^{k_{21}} x_2^{k_{22}} \dots x_n^{k_{2n}} + \dots + x_1^{k_{l1}} x_2^{k_{l2}} \dots x_n^{k_{ln}})^{p-1} = 1$$

is an identity. This can be checked in polynomial time by Lemma 148, hence the equation solvability problem for $\mathbf{G}$ is in P. □

6.3 Problems

Klíma's example mentioned in the beginning of the Chapter suggests the following question:

Problem 5. Is there an algebra $\mathbf{A}$ such that the polynomial equivalence problem for $\mathbf{A}$ is coNP-complete, but the equation solvability problem for $\mathbf{A}$ is in P?

If there is an example, it is not a group. Indeed, for a group $\mathbf{G}$ every instance $f_1 \approx f_2$ of the polynomial equivalence problem for $\mathbf{G}$ can be rewritten in the form $f_1 f_2^{-1} \approx 1$. If one can check the solvability of $p = a$ in polynomial time, then one only has to check the solvability of $f_1 f_2^{-1} = g$ for every $g \neq 1$. The two polynomials are equivalent if and only if none of these equations have a solution.

The smallest group not discussed in this Chapter is $\mathbf{S}_4$. This group can be considered as a semidirect product of $\mathbf{Z}_2^2$ and $\mathbf{S}_3$. Here, the exponent of the

first group is squarefree, the equivalence problem for $\mathbf{S}_3$ is in P, but the action of $\mathbf{S}_3$ is not Abelian. If we attack this problem using our technics, then after the collecting procedure, going along the lines of the proof of Theorem 151 or Theorem 154, we should discuss terms over $\mathbf{M}_2(\mathbf{Z}_2)$ evaluated on the invertible elements.

Problem 6. Find the complexity of the equivalence, the polynomial equivalence and the equation solvability problems for $\mathbf{S}_4$.

Chapter 7

The complexity of the equivalence problem for nonsolvable groups

In this Chapter we deal with non-solvable finite groups. A corollary of Theorem 146 is that the polynomial equivalence problem is coNP-complete for finite simple groups. In this Chapter we prove that this result is true for not only simple but for every non-solvable group and not only for the polynomial equivalence problem but for the equivalence problem:

Theorem 156. *The equivalence problem for a finite nonsolvable group* $\mathbf{G}$ *is coNP-complete.*

Let us recall a notation from Chapter 5. To every term expression $t(x_1, \ldots, x_n)$ and each group $\mathbf{G}$ we denote the naturally associated function by $t^{\mathbf{G}} \colon G^n \to G$. We recall that a group $\mathbf{G}$ *satisfies* an equation $s(\vec{x}) \approx t(\vec{x})$ for $\vec{x} = (x_1, \ldots, x_n)$, if the corresponding term functions $s^{\mathbf{G}}$ and $t^{\mathbf{G}}$ are the same function. We denote it by $\mathbf{G} \models s \approx t$. We recall that $\mathbf{G} \models s \approx t$ if and only if $\mathbf{G} \models s \cdot t^{-1} \approx 1$. Therefore we view the equivalence problem for groups as the problem of deciding which equations $t \approx 1$ are satisfied by $\mathbf{G}$.

Now we recall some definitions and easy observations about commutators and solvable groups (for more details see [31]).

Definition 157. a. The commutator $[x, y]$ is a group term defined by

$$[x, y] := x^{-1}y^{-1}xy.$$

b. Define the commutator terms $c_r\big(x_1, \ldots, x_{2^r}\big)$ by induction: $c_1(x_1, x_2) = [x_1, x_2]$ and for $r > 1$ let c_r be of arity 2^r:

$$c_r(x_1, x_2, \ldots, x_{2^r}) = \big[c_{r-1}(x_1, \ldots, x_{2^{r-1}}), c_{r-1}(x_{2^{r-1}+1}, \ldots, x_{2^r})\big].$$

c. $\mathbf{G}$ is *solvable* if and only if for some $r \geq 1$, $\mathbf{G} \models c_r \approx 1$. The smallest possible r is called the solvable length of $\mathbf{G}$.

d. For $a \in G$ let

$$[a, \mathbf{G}] \ := \ \big\langle \big\{ [a, g] : g \in G \big\} \big\rangle \, .$$

Lemma 158. *a. If $\mathbf{N} \trianglelefteq \mathbf{G}$ with both $\mathbf{N}$ and $\mathbf{G}/\mathbf{N}$ are solvable then $\mathbf{G}$ is also solvable.*

b. If $\mathbf{N}_1, \mathbf{N}_2$ are two normal solvable subgroups of $\mathbf{G}$ then the product $\mathbf{N}_1 \cdot \mathbf{N}_2$ is also a normal solvable subgroup of $\mathbf{G}$.

c. $[a, \mathbf{G}]$ is a normal subgroup of $\mathbf{G}$.

d. If $\mathbf{G}$ is a non-abelian simple group then

$$[a, \mathbf{G}] \ = \ \begin{cases} \mathbf{1} & \text{if } a = 1 \\ \mathbf{G} & \text{if } a \neq 1 \end{cases} .$$

Here are some notations and claims about the verbal subgroups of a group (see [28]).

Definition 159. a. Given a set T of group terms and let

$$T(\mathbf{G}) \ := \ \bigcup_{t \in T} \mathsf{Range}(t^{\mathbf{G}})$$

the union of the ranges of the term functions $t^{\mathbf{G}}$.

b. The subgroup generated by $T(\mathbf{G})$, which we denote by

$$T^*(\mathbf{G}) \ := \ \langle T(\mathbf{G}) \rangle$$

is called a *verbal* subgroup of $\mathbf{G}$.

c. $\mathbf{1}$ and $\mathbf{G}$ are verbal subgroups of $\mathbf{G}$. If these are the only verbal subgroups of $\mathbf{G}$ then we say $\mathbf{G}$ is *verbally simple*.

d. Given two terms $s(x_1, \ldots, x_m)$ and $t(x_1, \ldots, x_n)$, we define the term s_t by

$$s_t\big(x_1, \ldots, x_{mn}\big) \ := \ s\big(t(x_1, \ldots, x_n), t(x_{n+1}, \ldots, x_{2n}), \ldots, t(x_{mn-n+1}, \ldots, x_{mn})\big).$$

e. For a finite group $\mathbf{G}$ let $d_{\mathbf{G}}$ be a positive integer such that for any set X of generators of $\mathbf{G}$ we have

$$G = \bigcup_{0 \le k \le d_{\mathbf{G}}} X^k.$$

f. Given a term $s(x_1, \ldots, x_m)$ and a finite group $\mathbf{G}$ define the term $s_{\mathbf{G}}$ by

$$s_{\mathbf{G}}(x_1, \ldots, x_{md_{\mathbf{G}}}) := \underbrace{s(x_1, \ldots, x_m) \cdot s(x_{m+1}, \ldots, x_{2m}) \cdots}_{\text{a product of } d_{\mathbf{G}} \text{ terms } s(\cdots), \text{ with distinct variables}}.$$

Lemma 160. *a. Every verbal subgroup of $\mathbf{G}$ is normal in $\mathbf{G}$.*

b. A finite group $\mathbf{G}$ has a unique largest solvable verbal subgroup.

c. Suppose $\mathbf{G}$ is finite. If $T = \{t_1, \ldots, t_k\}$ let $t = t_1 \cdots t_k$. Then

$$T^*(\mathbf{G}) = t_{\mathbf{G}}(\mathbf{G}).$$

d. Thus for a finite $\mathbf{G}$, every verbal subgroup $\mathbf{V}$ of $\mathbf{G}$ is the range of a single term function.

The length of a term is important in our investigations.

Definition 161.

We recall that the length of a term function is defined inductively (by Definition 35): the length of a variable or its inverse is 1, and if s and t are terms with length a and b, then the length of the product term st is $a + b$.

Lemma 162. *a. The length of s_t is the product of the length of t and the length of s.*

b. The length of $s_{\mathbf{G}}$ is the product of $d_{\mathbf{G}}$ and the length of s.

The following proposition plays a crucial role in the proof of Theorem 156.

Proposition 163. *Let $\mathbf{G}$ be a finite group.*

a. For a verbal subgroup $\mathbf{V}$ let s be a term with $s(\mathbf{G}) = \mathbf{V}$. For all terms t we have

$$\mathbf{V} \models t \approx 1 \text{ if and only if } \mathbf{G} \models t_s \approx 1.$$

b. *Suppose* $\mathbf{G}$ *is nonsolvable but every proper verbal subgroup of* $\mathbf{G}$ *is solvable. Let* $\mathbf{V}$ *be the largest solvable verbal subgroup of* $\mathbf{G}$*, denote its solvable length by* r*. Then for all terms* t *we have*

$$\mathbf{G}/\mathbf{V} \models t \approx 1 \quad \textit{if and only if} \quad \mathbf{G} \models c_{rt_{\mathbf{G}}} \approx 1.$$

c. *If* $\mathbf{G}$ *is verbally simple and* $\mathbf{N}$ *is a proper normal subgroup of* $\mathbf{G}$ *then for all terms* t *we have*

$$\mathbf{G} \models t \approx 1 \quad \textit{if and only if} \quad \mathbf{G}/\mathbf{N} \models t \approx 1.$$

Proof. a. Let t be n-ary and s be m-ary. Let $\vec{y}_i = (y_{i1}, \dots, y_{im})$ for $i = 1, \dots, n$, and we consider the terms $t(x_1, \dots, x_n)$ and $t_s\big(y_{11}, \dots, y_{nm}\big) = t\big(s(\vec{y}_1), \dots, s(\vec{y}_n)\big)$. While $\vec{y}_i$ run through all tuples from $\mathbf{G}$, the values of $s(\vec{y}_i)$ attain every element of $\mathbf{V}$. Thus if $t \neq 1$ at some evaluation $(h_1, \dots, h_n) \in \mathbf{V}^n$, then we can choose the tuples $\vec{y}_i$ such that $s(\vec{y}_i) = h_i$. Thus there is an evaluation of t_s such that $t_s \neq 1$.

On the other hand, if $t_s \not\approx 1$ over $\mathbf{G}$, then there is an evaluation $\vec{y}_1, \dots, \vec{y}_k$ such that $t_s \neq 1$. Now, for the elements $h_i = s(\vec{y}_i)$ we have $t(h_1, \dots, h_n) \neq 1$, hence $t \not\approx 1$ over $\mathbf{V}$.

b. Let m be the arity of $t_{\mathbf{G}}$. If $t \approx 1$ over $\mathbf{G}/\mathbf{V}$, then $t_{\mathbf{G}}(\mathbf{G}) \leq \mathbf{V}$, hence $t_{\mathbf{G}}(\mathbf{G})$ is solvable and $c_{rt_{\mathbf{G}}} \approx 1$ over $\mathbf{G}$. On the other hand, if $t \not\approx 1$ over $\mathbf{G}/\mathbf{V}$ then $t_{\mathbf{G}}(\mathbf{G})$ is non-solvable and $t_{\mathbf{G}}(\mathbf{G}) = \mathbf{G}$. As there are some elements $g_1, \dots g_{2^r} \in \mathbf{G}$ such that $c_r(\vec{g}) \neq 1$, and there are m-tuples $\vec{y}^i$ such that $t_{\mathbf{G}}(\vec{y}^i) = g_i$, we have $c_{rt_{\mathbf{G}}}(\vec{y}^1, \dots, \vec{y}^{2^r}) \neq 1$. Hence $c_{rt_{\mathbf{G}}} \not\approx 1$ over $\mathbf{G}$.

c. If $t \approx 1$ over $\mathbf{G}$ then clearly $t \approx 1$ over $\mathbf{G}/\mathbf{N}$. Now, if $t \approx 1$ over $\mathbf{G}/\mathbf{N}$, then $t_{\mathbf{G}}(\mathbf{G}) \leq \mathbf{N}$. As $t_{\mathbf{G}}(\mathbf{G})$ is verbal, $t_{\mathbf{G}}(\mathbf{G}) = \mathbf{1}$, hence $t \approx 1$ over $\mathbf{G}$. □

7.1 Proving coNP-completeness

Our leading reference on computational complexity will be [7]. The equivalence problem of any finite group $\mathbf{G}$ is clearly in co-NP: to check if an equation $t(\vec{x}) \approx 1$ fails in $\mathbf{G}$ one only needs one instance $\vec{g}$ where $t^{\mathbf{G}}(\vec{g}) \neq 1$, and given such an instance $\vec{g}$ one can find the value of $t^{\mathbf{G}}(\vec{g})$ in polynomial time. Thus to prove the theorem we will exhibit an NP-complete problem that polynomially reduces to the equivalence problem of $\mathbf{G}$. The most elegant choice we have found is to use the NP-completeness of the k-coloring

problem where k is the size of the group $\mathbf{G}$ when $\mathbf{G}$ is a simple non-Abelian group. Then we use induction for non-solvable groups in general.

Theorem 164. *Let $\mathbf{G}$ be a finite, simple, non-Abelian group. Then the equivalence problem for $\mathbf{G}$ is coNP-complete.*

Proof. Let $k = |\mathbf{G}|$. The group $\mathbf{G}$ is non-Abelian and simple, hence $k \geq 60$. We polynomially reduce GRAPH k-COLORING to the equivalence problem of $\mathbf{G}$. Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We shall color the vertices of Γ by the elements of $\mathbf{G}$. The color of v_i will be g_i. We exhibit a term function t over $\mathbf{G}$ such that $t(g_1, \ldots, g_n) \neq 1$ if and only if the appropriate coloring is a k-coloring.

By Lemma 158/d we have $[g, \mathbf{G}] = \mathbf{G}$ for every $g \neq 1$. Let $d_{\mathbf{G}}$ be the constant defined in Definition 159/e. This constant is depending only on $\mathbf{G}$ and for every $g \in \mathbf{G}$

$$\mathbf{G} = [g, \mathbf{G}] = \prod_{1}^{d_{\mathbf{G}}} [g, y_i]$$

holds. Let

$$S(x, y_1, \ldots, y_{d_{\mathbf{G}}}) = S(x, \bar{y}) = \prod_{k=1}^{d_{\mathbf{G}}} [x, y_k].$$

Every vertex v_i in V will be associated to a variable x_i. Then for every edge $e = (v_i, v_j)$ we define

$$S_{i,j}(\bar{y}) = S(x_i x_j^{-1}, \bar{y}).$$

Thus $S_{i,j}(\mathbf{G}) = 1$ if we substitute $x_i = x_j$ and $S_{i,j}(\mathbf{G}) = \mathbf{G}$ if we substitute $x_i \neq x_j$. The length of $S_{i,j}$ depends only on $\mathbf{G}$: each commutator contains 3 variables, repeated twice and we multiply $d_{\mathbf{G}}$ of them, so the length of this term fuction is $6d_{\mathbf{G}}$. We are ready to define t. Let $e = (v_i, v_j)$ be an edge of Γ. Let

$$t_e(\bar{y}) = S_{i,j}(\bar{y}) = S(x_i x_j^{-1}, \bar{y}).$$

Let $e_1, e_2, \ldots, e_m$ be the list of edges of Γ and r such that $2^{r-1} < m \leq 2^r$. Moreover let

$$t = c_r(t_{e_1}, t_{e_2}, \ldots, t_{e_m}, t_{e_m}, \ldots, t_{e_m}).$$

Here we repeat t_{e_m} enough many ($2^r - m$ many) times in order to match the arity of c_r. In the terms t_{e_i} the variables of $\bar{y}$ are all distinct. So there are altogether $d_{\mathbf{G}} 2^r$ many 'y'-s and their inverses. The length of t is $6d_{\mathbf{G}} \cdot 4^r \leq 6d_{\mathbf{G}}(2m)^2 = 24 d_{\mathbf{G}} m^2$ hence polynomial in the size of Γ. We claim that $t \not\approx 1$ over $\mathbf{G}$ if and only if Γ is k-colorable. Firstly, let us assume that Γ is k-colorable by the elements of G, and let g_i be the color of v_i. Now,

substituting $x_i = g_i$, for every edge e of Γ we have $t_e(\mathbf{G}) = \mathbf{G}$. Since $\mathbf{G}$ is not solvable, $c_r \not\approx 1$ over $\mathbf{G}$ and so $t \not\approx 1$, either. Secondly, if $\mathbf{G}$ is not k-colorable, then at any assignment of the variables we have a monochromatic edge, e. Then $t_e = 1$ at every substitution, hence $t = 1$ at every substitution, thus $t \approx 1$. □

The first step of the induction is about verbal subgroups.

Lemma 165. *Let $\mathbf{V}$ be a verbal subgroup of $\mathbf{G}$. If the equivalence problem for $\mathbf{V}$ is coNP-complete, then the equivalence problem for $\mathbf{G}$ is coNP-complete.*

Proof. We give a polynomial reduction from the equivalence problem of $\mathbf{V}$ to the equivalence problem of $\mathbf{G}$. For every term function $t(x_1, \ldots, x_k)$ over $\mathbf{V}$ we present a term function t' over $\mathbf{G}$ such that $t \approx 1$ over $\mathbf{V}$ if and only if $t' \approx 1$ over $\mathbf{G}$. As $\mathbf{V}$ is verbal, there is a term $s(x_1, \ldots, x_n)$ over $\mathbf{G}$ such that $s(\mathbf{G}) = \mathbf{V}$. Let $t' = t_s$ as in Proposition 163/a. Now $t \approx 1$ over $\mathbf{V}$ if and only if $t' \approx 1$ over $\mathbf{G}$.

The reduction is polynomial in the length of t because the length of t' is the product of the length of t and the length of s. The latter depends only on the group $\mathbf{G}$. □

Now, we prove Theorem 156.

of Theorem 156 . We proceed by induction on the order of $\mathbf{G}$.

Case 1: There exists a non-trivial, non-solvable verbal subgroup $\mathbf{V}$ of $\mathbf{G}$. Now, $|\mathbf{V}| < |\mathbf{G}|$ and the equivalence problem for $\mathbf{V}$ is coNP-complete by the assumption. Thus the equivalence problem for $\mathbf{G}$ is coNP-complete by Lemma 165.

Case 2: There are no nontrivial nonsolvable verbal subgroups of $\mathbf{G}$ but there is a non-trivial solvable verbal subgroup of $\mathbf{G}$. Let $\mathbf{V}$ be the largest solvable verbal subgroup and r denote its solvable length. The quotient group $\mathbf{G}/\mathbf{V}$ is non-solvable. Now, the equivalence problem for $\mathbf{G}/\mathbf{V}$ is co-NP-complete by the assumption, as $|\mathbf{G}/\mathbf{V}| < |\mathbf{G}|$. We give a polynomial reduction from the equivalence problem for $\mathbf{G}/\mathbf{V}$ to the equivalence problem for $\mathbf{G}$.

Let t be a term over $\mathbf{G}/\mathbf{V}$. Then we know by Proposition 163/b that $t \approx 1$ over $\mathbf{G}/\mathbf{V}$ if and only if $c_{rt_\mathbf{G}} \approx 1$ over $\mathbf{G}$. The length of $c_{rt_\mathbf{G}}$ is the product of the length of c_r and the length of $t_\mathbf{G}$, which is the product of t and $d_\mathbf{G}$. The latter and the length of c_r depend only on the group $\mathbf{G}$, hence the reduction is polynomial.

Case 3: There are no verbal subgroups in $\mathbf{G}$. If $\mathbf{G}$ is simple, we are done by Theorem 164. Let $\mathbf{N}$ be a normal subgroup of $\mathbf{G}$ and t be a term function. By Proposition 163/c we know that $t \approx 1$ over $\mathbf{G}$ if and only

if $t \approx 1$ over $\mathbf{G}/\mathbf{N}$. The factor group $\mathbf{G}/\mathbf{N}$ is non-solvable, because $\mathbf{G}'$ is verbal and so $\mathbf{G}' = \mathbf{G}$. Thus by induction the equivalence problem for $\mathbf{G}$ is coNP-complete. □

7.2 Problems

There is still work left to be done if one wants to prove a result similar to Theorem 147.

Problem 7. Give an algebraic characterization of the class of finite groups with a polynomial time equivalence problem; likewise for the class of finite groups with a coNP-complete equivalence problem.

It is not yet clear whether or not these two complexity classes exhaust all finite groups.

Problem 8. Is there a polynomial time/coNP-complete dichotomy for the equivalence problem for finite groups?

Chapter 8

The extended equivalence problem for groups

In Section 3.6 we observed that the commutator as a basic operation can significantly change the length of realizing polynomials for several group-functions. For example, the expression $[[[x_1, x_2], x_3], \ldots, x_n]$ has length n if the commutator is a basic operation, but has exponential length in n when expressed by only the group multiplication. Such a decrease in the length suggests that the complexity of the equivalence problem might change if the commutator is taken as a basic operation. Other group operations might have a similar property. A straightforward question arises, whether the complexity of the equivalence problem changes by taking one or more new operations as additional basic operations. Moreover, this question is interesting not only for groups but for all finite algebras. Hence we can raise the question in general:

Definition 166. Let $\mathbf{A} = (A, g_1, \ldots, g_m)$ be a finite algebra with underlying set A and with basic operations $g_1, \ldots, g_m$. Let $f_1, \ldots, f_n$ be polynomial expressions over the algebra $\mathbf{A}$.
The algebra $(\mathbf{A}, f_1, \ldots, f_n)$ is defined to be the algebra $(A, g_1, \ldots, g_m, f_1, \ldots, f_n)$, i.e. the algebra with underlying set A and with basic operations $g_1, \ldots, g_m$ together with $f_1, \ldots, f_n$ as well.

1. The *extended equivalence problem* for $\mathbf{A}$.
   We say that the extended equivalence problem for $\mathbf{A}$ is in P if for all possible term expressions $f_1, \ldots, f_n$, built up from variables and the basic operations of $\mathbf{A}$, the equivalence problem over $(\mathbf{A}, f_1, \ldots, f_n)$ is in P.

   We say that the extended equivalence problem for $\mathbf{A}$ is coNP-complete

if there exist some term expressions $f_1, \ldots, f_n$, built up from variables and the basic operations of $\mathbf{A}$, such that the equivalence problem over $(\mathbf{A}, f_1, \ldots, f_n)$ is coNP-complete.

2. The *extended polynomial equivalence problem* for $\mathbf{A}$.
   We say that the extended polynomial equivalence problem for $\mathbf{A}$ is in P if for all polynomial expressions $f_1, \ldots, f_n$, built up from variables, constants from $\mathbf{A}$ and the basic operations of $\mathbf{A}$, the polynomial equivalence problem over $(\mathbf{A}, f_1, \ldots, f_n)$ is in P.

   We say that the extended polynomial equivalence problem for $\mathbf{A}$ is coNP-complete if there exist some polynomial expressions $f_1, \ldots, f_n$, built up from variables, constants from $\mathbf{A}$ and the basic operations of $\mathbf{A}$, such that the polynomial equivalence problem over $(\mathbf{A}, f_1, \ldots, f_n)$ is coNP-complete.

Remark 167. The extended equivalence problem is 'harder' than the (original) equivalence problem: by introducing new operations the length of a polynomial expression cannot increase and the complexity is determined by the length of the input expressions. Thus, if for an algebra $\mathbf{A}$ the equivalence problem is coNP-complete, then the extended equivalence problem for $\mathbf{A}$ is coNP-complete. If the extended equivalence problem for $\mathbf{A}$ is in P, then the (original) equivalence problem for $\mathbf{A}$ is in P. Similar statements can be derived for the polynomial equivalence and the extended polynomial equivalence problems. Moreover, the polynomial extended equivalence problem is 'harder' than the extended equivalence problem, since every term is a polynomial. Hence, if the extended equivalence problem is coNP-complete for $\mathbf{A}$, then the extended polynomial equivalence problem is coNP-complete for $\mathbf{A}$. If the extended polynomial equivalence problem is in P for $\mathbf{A}$, then the extended equivalence problem is in P for $\mathbf{A}$.

In this Chapter we consider the complexity of the extended equivalence problem and the extended polynomial equivalence problem for finite groups. We start with nilpotent groups in Section 8.1. The (original) equivalence and the polynomial equivalence problems for finite nilpotent groups are in P by Burris and Lawrence [3]. Using the idea of their proof we prove that the extended polynomial equivalence problem is in P.

Theorem 168. *Let $\mathbf{G}$ be a nilpotent finite group, let $f_1, f_2, \ldots, f_m$ be polynomial expressions built up from variables, constants of $\mathbf{G}$ and the basic operations of $\mathbf{G}$. Then the polynomial equivalence problem for $(\mathbf{G}, f_1, f_2, \ldots, f_m)$ is in P.*

We proved in Chapter 7 that for non-solvable groups the equivalence problem is coNP-complete. By Remark 167 we can conclude that the extended equivalence and the extended polynomial equivalence problems are coNP-complete for non-solvable groups. The complexity of the equivalence problem for non-nilpotent solvable groups is, for the most part, a *terra incognita* of mathematics. Only very few partial results are known (in Section 6.1 we proved that for a special class of meta-Abelian groups the complexity of the equivalence problem is in P, e.g. for meta-cyclic groups, dihedral groups $\mathbf{D}_{2k+1}$, $\mathbf{S}_3$ or $\mathbf{A}_4$), but we do not know the answer even for the symmetric group $\mathbf{S}_4$. The following theorem completes the characterization of the extended equivalence problem:

Theorem 169. *Let* $\mathbf{G}$ *be a finite solvable non-nilpotent group. Then there exists a term expression* f *(built up from variables and the basic operations of* $\mathbf{G}$*) such that the equivalence problem for* $(\mathbf{G}, f)$ *is coNP-complete.*

The function f is not uniform in these proofs; it depends on the group $\mathbf{G}$. However, we show in Section 8.5 that for a large class of groups f can be chosen as the *commutator*. From these results we immediately have the following corollary:

Corollary 170. *Let* $\mathbf{G}$ *be a finite group. If* $\mathbf{G}$ *is nilpotent then the extended equivalence and the extended polynomial equivalence problems are in P. If* $\mathbf{G}$ *is not nilpotent then the extended equivalence and the extended polynomial equivalence problems are coNP-complete.*

Comparing the results of Section 8.5 to the results of Section 6.1 we can conclude that the complexity of the equivalence and the extended equivalence problems are not always the same. By Theorem 151 the equivalence problem for $\mathbf{A}_4$ is in P. By Theorem 184 the equivalence problem for $(\mathbf{A}_4, [,])$ is coNP-complete.

8.1 Nilpotent groups

In [3] Burris and Lawrence state the following:

Proposition 171. *Let* $\mathbf{G}$ *be a finite nilpotent group with nilpotency class* c*. Let* $p(x_1, \ldots, x_n)$ *be a polynomial over* $\mathbf{G}$*. Then* $\mathbf{G} \models p(x_1, \ldots, x_n) \approx 1$ *if and only if* $p(a_1, \ldots, a_n) = 1$ *for every substitution* $(a_1, \ldots, a_n) \in G^n$*, where* $|\{ i : a_i \neq 1 \}| \leq c$*.*

This proposition claims that if one wants to check whether or not a polynomial p attains 1 for every substitution, then it is sufficient to check only

those substitutions where the value of at most c-many variables differ from 1. The following set contains all the necessary substitutions:

$$T = \{ (a_1, \ldots, a_n) \in G^n : |\{ i : a_i \neq 1 \}| \leq c \}.$$

Now $|T| = \sum_{i=0}^{c} \binom{n}{i} (|\mathbf{G}| - 1)^i \leq (c+1) |\mathbf{G}|^c \cdot n^c$, which is polynomial not only in the length of p but in the number of different variables of p as well. Finding T is polynomial in n, too. Checking, whether $p(a_1, \ldots, a_n) = 1$ for $(a_1, \ldots, a_n) \in T$ is polynomial in the length of p. Hence checking every substitutions from T requires polynomial time in n and in the length of p.

Proof of Theorem 168. Let $f_1, \ldots, f_k$ be polynomial expressions over $\mathbf{G}$ and let $p(x_1, \ldots, x_n)$ be a polynomial over $(\mathbf{G}, f_1, \ldots, f_k)$ (and not over $\mathbf{G}$). Let $p'(x_1, \ldots, x_n)$ be the polynomial we obtain after expanding p over $\mathbf{G}$, i.e. p' is a polynomial over $\mathbf{G}$ such that for every $(a_1, \ldots, a_n) \in \mathbf{G}^n$ we have $p(a_1, \ldots, a_n) = p'(a_1, \ldots, a_n)$. Now $(\mathbf{G}, f_1, \ldots, f_k) \models p \approx 1$ if and only if $\mathbf{G} \models p' \approx 1$. To decide whether or not $\mathbf{G} \models p' \approx 1$ we only have to check for every substitutions $(a_1, \ldots, a_n)$ from T, whether $p'(a_1, \ldots, a_n) = 1$. $p'(a_1, \ldots, a_n) = p(a_1, \ldots, a_n)$ and checking the value of $p(a_1, \ldots, a_n)$ is polynomial in the length of p. The number $|T|$ and finding the set T are both polynomial in n (so is in the length of p). Hence checking every substitutions from T requires polynomial time in n and in the length of p. □

Remark 172. Notice that the algorithm does not calculate p'. We only used p' for proving that $|T|$-many substitutions are sufficient to check whether or not $(\mathbf{G}, f_1, \ldots, f_k) \models p \approx 1$. The length of p' might not necessarily be polynomial in the length of p.

8.2 Preliminaries

First we list the necessary notations and definitions from group theory. We denote the commutator in a group $\mathbf{G}$ with $[,]$: $[x, y] = x^{-1}y^{-1}xy$. The lower central series for a group $\mathbf{G}$ is the following sequence of normal subgroups: $\gamma_0(\mathbf{G}) = \mathbf{G}$, $\gamma_i(\mathbf{G}) = [\mathbf{G}, \gamma_{i-1}(\mathbf{G})]$. It is clear that if $i < j$, then $\gamma_i(\mathbf{G}) \geq \gamma_j(\mathbf{G})$. For every finite group the lower central series terminates in $\gamma_{i_0}(\mathbf{G})$ for some i_0. Throughout this Chapter we denote this normal subgroup $\gamma_{i_0}(\mathbf{G})$ with $\mathbf{N} = \mathbf{N}(\mathbf{G})$. Recall that a group is nilpotent if and only if $\mathbf{N} = \mathbf{1}$. For a non-nilpotent, finite group the lower central series terminates in $\mathbf{N} \neq \mathbf{1}$. For a normal subgroup $\mathbf{H}$ of $\mathbf{G}$ and for every non-negative integer i we have $\gamma_i(\mathbf{G}/\mathbf{H}) = \gamma_i(\mathbf{G}) / (\mathbf{H} \cap \gamma_i(\mathbf{G}))$. Hence if $\mathbf{H}$ is a normal subgroup of a non-nilpotent finite group $\mathbf{G}$ such that $\mathbf{G}/\mathbf{H}$ is nilpotent, then $\mathbf{N} \leq \mathbf{H}$. The

following statement is an interesting structural theorem we use in the proof of Theorem 169:

Theorem 173. *Let* $\mathbf{G}$ *be a finite group. Let* $\mathbf{V}$ *be a normal subgroup of* $\mathbf{G}$ *such that* $\mathbf{G}' \leq \mathbf{V}$ *and both* $\mathbf{V}$ *and* $\mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$ *are nilpotent. Let* $\mathbf{N} = \mathbf{N}(\mathbf{G})$ *be as defined above. Then both* $\mathbf{N}$ *and* $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ *are Abelian.*

Proof. $\mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$ means that $\mathbf{N} \leq C_{\mathbf{G}}(\mathbf{V})$, and clearly $\mathbf{N} \leq \mathbf{G}' \leq \mathbf{V}$, hence $\mathbf{N} \leq C_{\mathbf{G}}(\mathbf{V}) \cap \mathbf{V} = C_{\mathbf{V}}(\mathbf{V}) = Z(\mathbf{V})$, thus $\mathbf{N}$ is Abelian.

Moreover from $\mathbf{N} \leq C_{\mathbf{G}}(\mathbf{V})$ we have $C_{\mathbf{G}}(\mathbf{N}) \geq C_{\mathbf{G}}(C_{\mathbf{G}}(\mathbf{V})) \geq \mathbf{V} \geq \mathbf{G}'$, so $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N}) \leq \mathbf{G}/\mathbf{G}'$, hence $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ is Abelian. □

Let us recall that a group element $g \in \mathbf{G}$ is called a left-Engel element if for every $h \in \mathbf{G}$ there is a positive integer k_h such that $[[[h, g], g] \dots g] = 1$ where the commutator is iterated k_h-many times. The set of left-Engel elements form $F(\mathbf{G})$, the Fitting subgroup (see [1]) which is by definition the maximal nilpotent normal subgroup in $\mathbf{G}$.

We prove Theorem 169 in Section 8.4. The following theorem is the key:

Theorem 174. *Let* $\mathbf{G}$ *be a non-nilpotent, finite group, let* $\mathbf{N} = \mathbf{N}(\mathbf{G})$ *be as defined above. Let us suppose that the groups* $\mathbf{N}$ *and* $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ *are both Abelian. Then there exists a term expression* f *(built up from variables and the basic operations of* $\mathbf{G}$*) such that the equivalence problem for* $(\mathbf{G}, f)$ *is coNP-complete.*

We prove Theorem 174 in Section 8.3. Before that we list the necessary notations and definitions from ring theory. Let $\mathbf{R}$ be a finite, commutative, non-nilpotent ring, let $\mathbf{J}(\mathbf{R})$ be its Jacobson radical. By the Wedderburn–Artin Theorem [17] we know that $\mathbf{R}/\mathbf{J}(\mathbf{R})$ is the direct sum of finite fields $\mathbf{F}_1, \dots \mathbf{F}_l$. Recall that for every finite ring there exist a positive integer e such that r^e is idempotent (i.e. $(r^e)^2 = r^e$) for every $r \in \mathbf{R}$ and $r^e = 0$ for every $r \in \mathbf{J}(\mathbf{R})$. If $\mathbf{R}$ is commutative, then $r^e = 0$ implies $r \in \mathbf{J}(\mathbf{R})$: $r^e = 0$ in $\mathbf{R}$ implies $(r + \mathbf{J}(\mathbf{R}))^e = r^e + \mathbf{J}(\mathbf{R}) = 0 + \mathbf{J}(\mathbf{R})$ in $\mathbf{R}/\mathbf{J}(\mathbf{R}) = \oplus_{i=1}^{l} \mathbf{F}_i$. This means that $(r + \mathbf{J}(\mathbf{R}))^e$ has 0 in each coordinate, and so has $r + \mathbf{J}(\mathbf{R})$. Hence $r \in \mathbf{J}(\mathbf{R})$. In other words the Jacobson radical of a commutative ring is exactly the set of nilpotent elements. This is not necessarily true for arbitrary rings, e.g. the ring $\mathbf{M}_k(\mathbf{F})$ (for $k \geq 2$) contains nilpotent elements, but $\mathbf{J}(\mathbf{M}_k(\mathbf{F})) = \mathbf{0}$.

8.3 Meta-nilpotent groups

We prove Theorem 174. Let $\mathbf{G}$ be a non-nilpotent finite group and let $\mathbf{N} = \mathbf{N}(\mathbf{G})$ be defined as in Section 8.2. Let us suppose that $\mathbf{N}$ and

$\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ are both Abelian. Let $\mathbf{A} = \mathbf{N}(\mathbf{G})$ throughout the Section. The group $\mathbf{G}$ acts on $\mathbf{A}$ by conjugation and the action is isomorphic to $\mathbf{B} = \mathbf{G}/C_{\mathbf{G}}(\mathbf{A})$. Let $\varphi\colon \mathbf{G} \to \mathbf{B}$ be the natural homomorphism. Every element of $\mathbf{B}$ acts as an automorphism of $\mathbf{A}$, in particular every element acts as an endomorphism. Since $\mathbf{B}$ is commutative, the actions of $\mathbf{B}$ generate a finite nontrivial commutative subring $\mathbf{R}(\mathbf{B})$ of End $\mathbf{A}$.

Let us examine the elements of $\mathbf{R}(\mathbf{B})$, the ring generated by $\mathbf{B}$. We write the action as an exponent: the image of $a \in \mathbf{A}$ at the action of $r \in \mathbf{R}(\mathbf{B})$ will be denoted by a^r. With these notations $x^1 = x$ and for every $b = \varphi(g)$ we have $x^b = x^{\varphi(g)} = g^{-1}xg$ thus $x^{b-1} = x^{\varphi(g)-1} = x^{-1+\varphi(g)} = x^{-1}g^{-1}xg = [x, g]$. Sometimes we omit φ from the exponent: by x^g we mean the conjugation with the group element g and write $x^g = g^{-1}xg$. Obviously $x^g = x^{\varphi(g)}$ for every $x \in \mathbf{G}$.

Let $\mathbf{C} = \{ b - 1 \mid b \in \mathbf{B} \}$. Let $\mathbf{R}(\mathbf{C}) \leq$ End $\mathbf{A}$ be the subring generated by the action of the commutator elements from $\mathbf{R}(\mathbf{B})$:

$$\mathbf{R}(\mathbf{C}) = \langle \mathbf{C} \rangle = \langle \varphi(g) - 1 \mid g \in \mathbf{G} \rangle .$$

Let $|\mathbf{B}| = |\mathbf{C}| = c$ and let $|\mathbf{R}(\mathbf{C})| = d$.

The idea of the proof is the following: for any ring expression t we have $a^t \approx 1$ for every $a \in \mathbf{A}$ if and only if End $\mathbf{A} \models t \approx 0$. This statement still holds if we replace End $\mathbf{A}$ by any subring of End $\mathbf{A}$. It is coNP-complete to decide over a non-nilpotent commutative ring $\mathbf{R}$ whether or not $\mathbf{R} \models t \approx 0$ (see [16]). Hence if we choose a commutative non-nilpotent subring $\mathbf{R}$ of End $\mathbf{A}$ and we are able to translate the ring operations into group operations, then we can reduce the equivalence problem over $\mathbf{R}$ to the equivalence problem over $\mathbf{G}$. This subring needs to be verbal: there must exist an integer coefficient polynomial p such that if the variables of p run through over $\varphi(\mathbf{G})$ then $p(\mathbf{G})$ runs through on the elements of the subring $\mathbf{R}$. In our case $\mathbf{R}(\mathbf{C})$ plays the role of $\mathbf{R}$ as Lemma 175 and Lemma 176 show.

Unfortunately we cannot translate the ring operations over the group $\mathbf{G}$, we need to understand properly the structure of $\mathbf{R}(\mathbf{C})$ and follow a proof for the coNP-completeness of the equivalence problem over $\mathbf{R}(\mathbf{C})$. From Lemma 176 we know that $\mathbf{R}(\mathbf{C})$ is commutative and non-nilpotent, hence $\mathbf{R}(\mathbf{C})/\mathbf{J}(\mathbf{R}(\mathbf{C}))$ is the direct sum of finite fields $\mathbf{F}_1, \ldots, \mathbf{F}_l$. Let $q = \max_{1 \leq i \leq l} |\mathbf{F}_i|$. Lemma 177 tells us that $q > 2$.

After we understand the structure of $\mathbf{R}(\mathbf{C})$, we reduce the GRAPH q-COLORING problem to the equivalence problem over $\mathbf{G}$ in the following way: Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{ v_1, \ldots, v_n \}$ and $E = \{ e_1, \ldots, e_m \}$. With the help of Lemmas 178, 179

and 180 we exhibit a word t_Γ over $\mathbf{R}(\mathbf{C})$ such that $\mathbf{R}(\mathbf{C}) \models t_\Gamma \approx 0$ if and only if Γ is *not* q-colorable. For every graph Γ we exhibit a word $Q_\Gamma = a^{t_\Gamma}$ over $\mathbf{G}$ and Lemma 181 proves that $\mathbf{G} \models Q_\Gamma \approx 1$ if and only if Γ is *not* q-colorable. This finishes the reduction.

We observe though, that this reduction is not polynomial, since Q_Γ is exponentially long in the size of Γ when expressed using only the multiplication and the inverse operations of $\mathbf{G}$. Nevertheless there exists a term operation f (built up from variables and the basic operations of $\mathbf{G}$) such that using f makes Q_Γ polynomially long in the size of Γ, i.e. the length $\|Q_\Gamma\|_{(\mathbf{G},f)}$ is polynomial in n (the number of vertices in Γ) and in m (the number of edges in Γ).

Therefore the proof consists of the following steps:

1. In Lemma 175 we prove that $\mathbf{R}(\mathbf{C})$ is verbal.

2. In Lemma 176 we prove that $\mathbf{R}(\mathbf{C})$ is not nilpotent. Thus the factor $\mathbf{R}(\mathbf{C})/\mathbf{J}(\mathbf{R}(\mathbf{C}))$ is the direct sum of finite fields $\mathbf{F}_1, \ldots, \mathbf{F}_l$.

3. Let $q = \max_{1 \leq i \leq l} |\mathbf{F}_i|$. Lemma 177 tells us that $q > 2$. Thus the GRAPH q-COLORING problem is NP-complete.

4. Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{ v_1, \ldots, v_n \}$ and $E = \{ e_1, \ldots, e_m \}$. We exhibit a word t_Γ over $\mathbf{R}(\mathbf{C})$ such that $\mathbf{R}(\mathbf{C}) \models t_\Gamma \approx 0$ if and only if Γ is *not* q-colorable (Lemmas 178, 179 and 180).

5. We present a term expression f over $\mathbf{G}$. For every graph Γ we exhibit a word $Q_\Gamma = a^{t_\Gamma}$ over $(\mathbf{G}, f)$ and Lemma 181 proves that $(\mathbf{G}, f) \models Q_\Gamma \approx 1$ if and only if Γ is *not* q-colorable.

6. We prove that the length of Q_Γ over $(\mathbf{G}, f)$ is polynomial in the size of Γ. Thus we polynomially reduced the GRAPH q-COLORING problem (for some $q > 2$) to the equivalence problem over $(\mathbf{G}, f)$.

We start first with step 1 of the proof. Let us recall that $c = |\mathbf{B}| = |\mathbf{C}|$ and $d = |\mathbf{R}(\mathbf{C})|$.

Lemma 175. *There exists an integer coefficient polynomial p of cd-many variables such that* $\mathbf{R}(\mathbf{C}) = p(\mathbf{B}^{cd}) = p\left(\varphi\left(\mathbf{G}^{cd}\right)\right)$.

Proof. Obviously, $\mathbf{R}(\mathbf{C})$ consists of all (integer coefficient) polynomials of the elements $b_i - 1$ where b_i runs over the group $\mathbf{B}$. As $\mathbf{R}(\mathbf{B})$ is finite $\mathbf{R}(\mathbf{C})$ is finite, too. Let $r_1, r_2, \ldots, r_d$ denote the elements of $\mathbf{R}(\mathbf{C})$. For every $r_i \in \mathbf{R}(\mathbf{C})$ there is a polynomial $p_i \in \mathbb{Z}[x_1, x_2 \ldots, x_c]$ such that

$$r_i = p_i(b_1 - 1, b_2 - 1, \ldots, b_c - 1).$$

There are several polynomials of this form, we fix one for every i for the remaining of the proof. Let

$$p(\bar{x}) = \sum_{1 \leq i \leq d} p_i(x_{1,i} - 1, x_{2,i} - 1, \ldots, x_{c,i} - 1),$$

where all the variables $x_{j,i}$ differ from each other. We have $p(\mathbf{B}^{cd}) \subseteq \mathbf{R}(\mathbf{C})$ and substituting $x_{j,l} = 1$ for $l \neq i$ and $x_{j,i} = b_j$, we have that $r_i \in p(\mathbf{B}^{cd})$. Hence $\mathbf{R}(\mathbf{C}) = p(\mathbf{B}^{cd}) = p\left(\varphi\left(\mathbf{G}^{cd}\right)\right)$. □

We continue with step 2 of the proof.

Lemma 176. *The ring* $\mathbf{R}(\mathbf{C})$ *is not nilpotent.*

Proof. It is enough to show that there exists a $g \in \mathbf{G}$ such that $\varphi(g) - 1$ is not nilpotent in $\mathbf{R}(\mathbf{C})$. The element $\varphi(g) - 1$ is nilpotent if there exists some k such that $(\varphi(g) - 1)^k = 0$. For a group element $h \in \mathbf{G}$ we have $h^{\varphi(g)-1} = [h, g]$. Moreover $h^{(\varphi(g)-1)^k} = [[[h, g], g] \ldots g]$, where the commutator is iterated k-many times. Let us recall that a group element g is called a left-Engel element if for every $h \in \mathbf{G}$ there is a positive integer k_h such that $[[[h, g], g] \ldots g] = 1$, where the commutator is iterated k_h-many times. The set of left-Engel elements form $F(\mathbf{G})$, the Fitting subgroup. The Fitting subgroup is the maximal nilpotent normal subgroup in $\mathbf{G}$ (see [1]). By our assumption $F(\mathbf{G}) \neq \mathbf{G}$. Hence every $g \notin F(\mathbf{G})$ is not an Engel element and we can choose $h \in \mathbf{G}$ such that $[[[h, g], g] \ldots g]$ never terminates in the identity element. Moreover, if the commutator action of g is not nilpotent on $\mathbf{G}$, then it is not nilpotent on $\mathbf{N}(\mathbf{G})$, as for large enough k the element $h^{\varphi(g)-1} \in \mathbf{N}(\mathbf{G})$. As $\mathbf{A} = \mathbf{N}(\mathbf{G})$ throughout this Section, we have that $\varphi(g) - 1$ is not nilpotent for any $g \notin F(\mathbf{G})$, thus $\mathbf{R}(\mathbf{C})$ is not nilpotent. □

Lemma 176 implies that $\mathbf{R}(\mathbf{C})/\mathbf{J}(\mathbf{R}(\mathbf{C}))$ is the direct sum of finite fields $\mathbf{F}_1, \ldots, \mathbf{F}_l$. For every commutative ring there exists a positive natural number e such that $(r^e)^2 = r^e$ for every $r \in \mathbf{R}(\mathbf{C})$ and $r^e = 0$ if and only if $r \in \mathbf{J}(\mathbf{R}(\mathbf{C}))$. Let us fix an e with this property for this Section. We continue with step 3 of the proof.

Lemma 177. *For the ring* $\mathbf{R}(\mathbf{C})$ *we have* $\mathbf{R}(\mathbf{C})/\mathbf{J}(\mathbf{R}(\mathbf{C})) \neq \mathbf{Z}_2^n$.

Proof. If $\mathbf{R}(\mathbf{C})/\mathbf{J}(\mathbf{R}(\mathbf{C})) = \mathbf{Z}_2^n$ for some n, then $r^2+r \in \mathbf{J}(\mathbf{R}(\mathbf{C}))$ for every $r \in \mathbf{R}(\mathbf{C})$. Let e' be a natural number, such that the exponent of $\mathbf{G}$ divides e' and $e' \geq e$. Since $e' \geq e$, if $r' \in \mathbf{J}(\mathbf{R}(\mathbf{C}))$ then $(r')^{e'} = 0$. Substituting $r = b-1$ for any $b = \varphi(g)$ we have $\left((b-1)^2 + (b-1)\right)^{e'} = 0$ for every $b \in \mathbf{B}$. As $\mathbf{R}(\mathbf{C})$ is a commutative subring of the commutative ring $\mathbf{R}(\mathbf{B})$, the equation $\left((b-1)^2 + (b-1)\right)^{e'} = 0$ holds in $\mathbf{R}(\mathbf{B})$ as well. Now

$$\begin{aligned} 0 &= ((b-1)^2 + (b-1))^{e'} = ((b-1)\cdot(b-1+1))^{e'} \\ &= ((b-1)\cdot b)^{e'} = (b-1)^{e'} \cdot b^{e'} \\ &= (b-1)^{e'}. \end{aligned}$$

The equality $(b-1)^2 + (b-1) = (b-1)\cdot(b-1+1)$ holds because $1, b \in \mathbf{R}(\mathbf{B})$. The following equality holds as $b = b-1+1$. Again, as $\mathbf{R}(\mathbf{B})$ is commutative, we have $((b-1)\cdot b)^{e'} = (b-1)^{e'} \cdot b^{e'}$. Finally $b^{e'} = 1$, since the exponent of $\mathbf{G}$ divides e' and $\mathbf{G} \models g^{e'} \approx 1$. Now $0 = (b-1)^{e'} = (\varphi(g)-1)^{e'}$ means that commuting with the element g is a nilpotent action, which is not true for every $g \in \mathbf{G}$. The contradiction proves the lemma. □

Now we move on to step 4 of the proof. Let $q = \max_{1\leq i\leq l} |\mathbf{F}_i|$. We now give the polynomial reduction from GRAPH q-COLORING to the equivalence problem over $(\mathbf{G}, f)$ for a particular function f. By Lemma 177 we have $q \geq 3$, therefore the GRAPH q-COLORING is NP-complete. Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Let

$$\begin{aligned} t'_\Gamma(z_1, \ldots, z_n) &= \prod_{v_iv_j \in E} (z_i - z_j), \text{ and let} \\ t_\Gamma(z_1, \ldots, z_n) &= (t_\Gamma(z_1, \ldots, z_n))^e = \prod_{v_iv_j \in E} (z_i - z_j)^e. \end{aligned}$$

Lemma 178. *Let q' be a prime power. The graph Γ is* not *q'-colorable if and only if $\mathbf{GF}(q') \models t'_\Gamma \approx 0$.*

Proof. We color the vertices of Γ by the elements of $\mathbf{GF}(q')$. The color of v_i will be s_i. We prove that $t(s_1, \ldots, s_n) \neq 0$ if and only if the appropriate coloring is a q'-coloring of Γ.

First, let us assume that Γ is q'-colorable by the elements of $\mathbf{GF}(q')$, and let s_i be the color of v_i. Now, substituting $z_i = s_i$, for every edge $e = v_iv_j$ of Γ we have $z_i - z_j \neq 0$, hence $t \not\approx 0$. Conversely, if Γ is not q'-colorable, then at any assignment of the variables we have a monochromatic edge, $e = v_iv_j$. Then $t'_\Gamma = 0$ at every substitution. □

Lemma 179. *The graph* Γ *is* not q*-colorable if and only if* $\oplus_{i=1}^{l}\mathbf{F}_i \models t'_\Gamma \approx 0$.

Proof. By Lemma 178, if Γ is q-colorable, then $\mathbf{GF}(q) \models t'_\Gamma \not\approx 0$, hence $\oplus_{i=1}^{l}\mathbf{F}_i \models t'_\Gamma \not\approx 0$. If Γ is not q-colorable, then it is not q'-colorable for any $q' \leq q$, thus we have that $\mathbf{GF}(q') \models t'_\Gamma \approx 0$ for every $q' \leq q$. Hence $\mathbf{F}_i \models t'_\Gamma \approx 0$ and $\oplus_{i=1}^{l}\mathbf{F}_i \models t'_\Gamma \approx 0$. □

Lemma 180. *The graph* Γ *is* not q*-colorable if and only if* $\mathbf{R}(\mathbf{C}) \models t_\Gamma \approx 0$.

Proof. We chose the number e such that for every ring element $r \in \mathbf{R}(\mathbf{C})$ we have $r^e = 0$ in $\mathbf{R}(\mathbf{C})$ if and only if $r \in \mathbf{J}(\mathbf{R}(\mathbf{C}))$. Now $t_\Gamma = (t'_\Gamma)^e$, hence $\mathbf{R}(\mathbf{C}) \models t_\Gamma \approx 0$ if and only if $\mathbf{R}(\mathbf{C})/J(\mathbf{R}(\mathbf{C})) \models t'_\Gamma \approx 0$. Since $\mathbf{R}(\mathbf{C})/J(\mathbf{R}(\mathbf{C})) = \oplus_{i=1}^{l}\mathbf{F}_i$, Lemma 179 finishes the proof. □

An immediate consequence of Lemma 180 is that $\mathbf{G} \models a^{t_\Gamma(z_1,\ldots,z_n)} \not\approx 1$ (where a runs through $\mathbf{A}$ and z_i's run through $\mathbf{R}(\mathbf{C})$) if and only if Γ is q-colorable. We give an expression Q over $\mathbf{G}$ (using a new operation f built up from variables and the basic operations of $\mathbf{G}$) such that the image of Q over $\mathbf{G}$ will be the same as the image of $a^{t_\Gamma(z_1,\ldots,z_n)}$.

We continue with step 5 of the proof. Let us introduce a new operation f over the group $\mathbf{G}$ on $2cd+1$-many variables.

$$f(y, \bar{x}_1, \bar{x}_2) = y^{(z_1 - z_2)} = y^{(p(\bar{x}_1) - p(\bar{x}_2))},$$

where $z_i = p(\bar{x}_i)$, $x^{y+z} = x^y x^z = y^{-1}xyz^{-1}xz$, $x^{-y} = (x^{-1})^y = y^{-1}x^{-1}y$ and $x^{yz} = (x^y)^z = (yz)^{-1}xyz$.

Now we polynomially reduce the GRAPH q-COLORING problem to the equivalence problem over $(\mathbf{G}, f)$. Let $\Gamma = (V, E)$ be a graph, with n vertices, $V = \{v_1, \ldots v_n\}$ and m edges, $E = \{e_1, \ldots e_m\}$. Let $\bar{x}_1, \ldots, \bar{x}_n$ be different vectors of cd-many variables assigned to the vertices (we remind the reader that polynomial p is of cd-many variables). So there are altogether $n \cdot cd$ many 'x' variables and their inverses. Let us denote all these 'x' variables by $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$. We exhibit the expressions $y^{t'_\Gamma(z_1,\ldots,z_n)}$ and $y^{t_\Gamma(z_1,\ldots,z_n)}$. in the following way: for every edge e_i let $e_i = v_{i,1}v_{i,2}$ and let

$$\begin{aligned} w_1(y, \bar{x}) &= f(y, \bar{x}_{1,1}, \bar{x}_{1,2}), \\ w_i(y, \bar{x}) &= f \circ w_{i-1} = f\left(w_{i-1}(y, \bar{x}), \bar{x}_{i,1}, \bar{x}_{i,2}\right), \end{aligned}$$

where $\bar{x}_{i,j}$ is the vector of variables assigned to the vertex $v_{i,j}$. Let us denote

$p(\bar{x}_i)$ by z_i. Now it is easy to see, that

$$\begin{aligned} w_1(y,\bar{x}) &= y^{z_{1,1}-z_{1,2}}, \\ w_i(y,\bar{x}) &= w_{i-1}(y,\bar{x})^{z_{i,1}-z_{i,2}} \\ &= y^{(z_{1,1}-z_{1,2})\ldots(z_{i,1}-z_{i,2})}, \\ w_i(y,\bar{x}) &= y^{(z_{1,1}-z_{1,2})\ldots(z_{m,1}-z_{m,2})} \\ &= y^{\prod_{v_iv_j\in E}(z_i-z_j)} = y^{t'_\Gamma(z_1,\ldots,z_n)}. \end{aligned}$$

Now we exhibit the term expression $y^{t_\Gamma(z_1,\ldots,z_n)}$ by applying $t_\Gamma(z_1,\ldots,z_n) = (t'_\Gamma(z_1,\ldots,z_n))^e$. Let

$$\begin{aligned} W_1(y,\bar{x}) &= w_m(y,\bar{x}), \\ W_i(y,\bar{x}) &= W_1 \circ W_{i-1} = W_1(W_{i-1}(y,\bar{x}),\bar{x}). \end{aligned}$$

Now it is easy to see that

$$\begin{aligned} W_1(y,\bar{x}) &= y^{t'_\Gamma(z_1,\ldots,z_n)}, \\ W_i(y,\bar{x}) &= W_{i-1}(y,\bar{x})^{t'_\Gamma(z_1,\ldots,z_n)} \\ &= y^{\left(t'_\Gamma(z_1,\ldots,z_n)\right)^i}, \\ W_e(y,\bar{x}) &= y^{\left(t'_\Gamma(z_1,\ldots,z_n)\right)^e} \\ &= y^{t_\Gamma(z_1,\ldots,z_n)}. \end{aligned}$$

Now $\mathbf{A} = \mathbf{N}$ is a verbal subgroup of $\mathbf{G}$, let $W_0(\bar{y})$ be a word with image $\mathbf{A}$. We are interested in $Q_\Gamma = W_e(W_0(\bar{y}),\bar{x})$, where e was the natural number for which $(r^e)^2 = r^e$ for every $r \in \mathbf{R}(\mathbf{C})$ and $r^e = 0$ if and only if $r \in \mathbf{J}(\mathbf{R}(\mathbf{C}))$. Observe, that $Q_\Gamma = W_0(\bar{y})^{t_\Gamma(z_1,\ldots,z_n)}$ with the notation $z_i = p(\bar{x}_i)$.

Lemma 181. *The graph Γ is* not *q-colorable if and only if $(\mathbf{G},f) \models Q_\Gamma \approx 1$.*

Proof. If Γ is q-colorable, then $\mathbf{R}(\mathbf{C}) \models t_\Gamma \not\approx 0$, hence there exists a substitution of $z_1,\ldots,z_n$ from $\mathbf{R}(\mathbf{C})$ such that $t_\Gamma(z_1,\ldots,z_n) \neq 0$ over $\mathbf{R}(\mathbf{C})$. The image of the polynomial p over $\mathbf{B}$ is $\mathbf{R}(\mathbf{C})$, hence we can choose the tuples $\bar{x}_1,\ldots,\bar{x}_n$ from $\mathbf{G}$ such that $p(\varphi(\bar{x}_i)) = z_i$. With this evaluation $t_\Gamma \neq 0$ over $\mathbf{R}(\mathbf{C})$, hence there exists an $a \in \mathbf{A}$ such that $a^{t_\Gamma} \neq 1$ over $\mathbf{G}$. Let us choose $\bar{y}$ such that $a = W_0(\bar{y})$ and with this evaluation of the variables we have that $(\mathbf{G},f) \models Q_\Gamma \not\approx 1$. If Γ is not q-colorable, then we have that $\mathbf{R}(\mathbf{C}) \models t_\Gamma \approx 0$. Thus for every $a \in \mathbf{A}$ (especially $a = W_0(\bar{y})$) we have $a^{t_\Gamma} = 1$ and $(\mathbf{G},f) \models Q_\Gamma \approx 1$. □

Finally we finish with step 6 of the proof. Let us denote the length of an expression w with $\|w\|$. The reduction from GRAPH q-COLORING to the equivalence problem over $(\mathbf{G}, f)$ is polynomial, because the length of $Q_\Gamma = Q_e$ is $\|Q_\Gamma\| = \|Q_e\| \leq \|f\| \cdot m \cdot e \cdot (ncd + \|W_0\|)$: when building up Q_Γ we use the function f exactly $e \cdot m$-many times on first input of length $\|W_0\|$ and on ncd-many variables. Hence $\|Q_\Gamma\|$ is polynomial in the size of Γ and Theorem 174 is proved.

8.4 Non-nilpotent groups

First we prove two lemmas which play a great role in the inductive proof of Theorem 169.

Lemma 182. *Let* $\mathbf{H}$ *be a verbal subgroup of* $\mathbf{G}$ *and let* f *be a term operation (built up from variables and from the basic group operations). If the equivalence problem for* $(\mathbf{H}, f)$ *is coNP-complete, then the equivalence problem for* $(\mathbf{G}, f)$ *is coNP-complete, too.*

Proof. We give a polynomial reduction from the equivalnce problem for $(\mathbf{H}, f)$ to the equivalence problem for $(\mathbf{G}, f)$.

For every word $w(x_1, \ldots, x_n)$ over $(\mathbf{H}, f)$ we present a word w' over $(\mathbf{G}, f)$ such that $(\mathbf{H}, f) \models w \approx 1$ if and only if $(\mathbf{G}, f) \models w' \approx 1$. As $\mathbf{H}$ is verbal, there is a word $v(x_1, \ldots, x_k)$ over $\mathbf{G}$ such that the image of v over $\mathbf{G}$ is $\mathbf{H}$. Let w' be the composition of w and v: substitute v into every variable x_i of w. Let $\bar{y}_i = (y_{i1}, \ldots, y_{ik})$ for $i = 1, \ldots, n$ and let

$$w'(\bar{y}_1, \ldots, \bar{y}_n) = w(v(\bar{y}_1), \ldots, v(\bar{y}_n)).$$

While $\bar{y}_i$ runs through all tuples from $\mathbf{G}$, the values of $v(\bar{y}_i)$ attain every element of $\mathbf{H}$. Thus if $w \neq 1$ at some evaluation $(h_1, \ldots, h_n) \in \mathbf{H}^n$, then we can choose the tuples $\bar{y}_i$ such that $t(\bar{y}_i) = h_i$. Thus there is an evaluation of w' such that $w' \neq 1$.

On the other hand, if $(\mathbf{G}, f) \models w' \not\approx 1$, then there is an evaluation $\bar{y}_1, \ldots, \bar{y}_n$ such that $w' \neq 1$. Now, for the elements $h_i = v(\bar{y}_i)$ we have $w(h_1, \ldots, h_n) \neq 1$, hence $(\mathbf{H}, f) \models w \not\approx 1$.

The reduction is polynomial in the length of w because the length of w' is at most the product of the length of w and the length of v (we changed every variable to v). The latter depends only on the group $\mathbf{G}$. □

Lemma 183. *Let* $\mathbf{V}$ *be a verbal subgroup of* $\mathbf{G}$ *and let* $\mathbf{H} = \mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$. *Let* f *be a term operation (built up from variables and from the basic group operations). If the equivalence problem for* $(\mathbf{H}, f)$ *is coNP-complete, then the equivalence problem for* $(\mathbf{G}, f)$ *is coNP-complete, too.*

Proof. As $\mathbf{V}$ is verbal, there is a word $v(y_1, \ldots, y_k)$ over $\mathbf{G}$ such that the image of v over $\mathbf{G}$ is $\mathbf{V}$. Let $\bar{y} = (y_1, \ldots, y_k)$. We give a polynomial reduction from the equivalence problem for $(\mathbf{H}, f)$ to the equivalence problem for $(\mathbf{G}, f)$. If we need to check whether or not $(\mathbf{H}, f) \models w(x_1, \ldots, x_n) \approx 1$, then we consider the word

$$\begin{aligned} w' &= (w(x_1, \ldots, x_n))^{-1} (v(\bar{y}))^{-1} w(x_1, \ldots, x_n) v(\bar{y}) \\ &= [w(x_1, \ldots, x_n), v(\bar{y})] \end{aligned}$$

over $(\mathbf{G}, f)$. We prove that $(\mathbf{G}, f) \models w' \approx 1$ if and only if $(\mathbf{H}, f) \models w \approx 1$. First, if $(\mathbf{H}, f) \models w(x_1, \ldots, x_n) \approx 1$, then $w(x_1, \ldots, x_n) \in C_{\mathbf{G}}(\mathbf{V})$ if we substitute from $\mathbf{G}$. Thus commuting it with any $y_0 = v(\bar{y}) \in \mathbf{V}$ we have $(\mathbf{G}, f) \models [w(x_1, \ldots, x_n), y_0] \approx 1$. Conversely, if $(\mathbf{G}, f) \models [w(x_1, \ldots, x_n), v(\bar{y})]$, then $w(x_1, \ldots, x_n) \in C_{\mathbf{G}}(\mathbf{V})$ for every substitution over $\mathbf{G}$, hence $(\mathbf{H}, f) \models w(x_1, \ldots, x_n) \approx 1$. The reduction is polynomial, because the length of w' is at most twice as the sum of the length of w and the length of v. □

Proof of Theorem 169. We proceed by induction on the order of $\mathbf{G}$. Let $\mathbf{V}$ be any verbal normal subgroup with the property $\mathbf{G} \neq \mathbf{V} \geq \mathbf{G}'$. Such a verbal subgroup exists, e.g. $\mathbf{V} = \mathbf{G}' \leq \mathbf{G}$ as $\mathbf{G}$ is solvable. Let us fix such a $\mathbf{V}$ for the proof.

Case 1: $\mathbf{V}$ is not nilpotent. Now $|\mathbf{V}| < |\mathbf{G}|$ and by the assumption there exists a function f (built up from variable and from basic group operations) such that the equivalence problem for $(\mathbf{H}, f)$ is coNP-complete. Thus the equivalence problem for $(\mathbf{G}, f)$ is coNP-complete by Lemma 182.

Case 2: $\mathbf{V}$ is nilpotent but $\mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$ is not nilpotent. Let $\mathbf{H} = \mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$. Since $\mathbf{V}$ is nilpotent $1 \neq Z(\mathbf{V}) \leq C_{\mathbf{G}}(\mathbf{V})$ and $|\mathbf{H}| < |\mathbf{G}|$. The group $\mathbf{H}$ is not nilpotent, hence there exists a function f (built up from variable and from basic group operations) such that the equivalence problem for $(\mathbf{H}, f)$ is coNP-complete and so is the equivalence problem for $(\mathbf{G}, f)$ by Lemma 183.

Case 3: $\mathbf{V}$ and $\mathbf{G}/C_{\mathbf{G}}(\mathbf{V})$ are both nilpotent. Let $\mathbf{N} = \mathbf{N}(\mathbf{G})$ be as defined in Section 8.2. By Theorem 173 we have that both $\mathbf{N}$ and $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ are Abelian. Theorem 174 finishes the proof. □

8.5 Choosing the commutator

With a deeper analysis of the structure of non-nilpotent groups, we can prove that the commutator is usually enough to obtain coNP-complete extended equivalence problem.

Theorem 184. *Let* $\mathbf{G}$ *be a non-nilpotent group, let* $\mathbf{N} = \mathbf{N}(\mathbf{G})$ *be as defined in Section 8.2. Let us suppose that* $\mathbf{G}/C_{\mathbf{G}}(\mathbf{N})$ *and* $\mathbf{N}$ *are both Abelian. Let us suppose that* $\exp(\mathbf{G}/F(\mathbf{G})) > 2$*, where* $F(\mathbf{G})$ *is the Fitting subgroup of the group* $\mathbf{G}$*. Then the equivalence problem for* $(\mathbf{G}, [,])$ *is coNP-complete, where* $[,]$ *denotes the commutator operation.*

Corollary 185. *The equivalence problem for* $(\mathbf{A}_4, [,])$ *is coNP-complete.*

We use similar notations as in Section 8.3. Let $\mathbf{A} = \mathbf{N}$. The group $\mathbf{G}$ acts on $\mathbf{A}$ by conjugation and the action is isomorphic to $\mathbf{B} = \mathbf{G}/C_{\mathbf{G}}(\mathbf{A})$. Let $\varphi\colon \mathbf{G} \to \mathbf{B}$ be the natural homomorphism. Similarly as in Section 8.3, every element of $\mathbf{B}$ acts as an automorphism of $\mathbf{A}$, in particular every element acts as an endomorphism. Since $\mathbf{B}$ is commutative, the actions of $\mathbf{B}$ generate a finite nontrivial commutative subring $\mathbf{R}(\mathbf{B})$ of End $\mathbf{A}$. Since $\mathbf{B}$ is commutative, it generates a finite nontrivial commutative unitary subring $\mathbf{R}(\mathbf{B})$ of End $\mathbf{A}$. Let $\exp(\mathbf{G}/F(\mathbf{G})) = q \geq 3$. $\mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$ is a sum of finite fields $\mathbf{F}_1, \ldots, \mathbf{F}_k$. Let e_0 be a positive natural number such that $(r^{e_0})^2 = r^{e_0}$ for every $r \in \mathbf{R}(\mathbf{B})$ and $r^{e_0} = 0$ if and only if $r \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$.

First we prove three structural lemmas about $\mathbf{R}(\mathbf{B})$ (Lemmas 186, 187 and 188), then we move on to the proof of Theorem 184. Let us recall that the Fitting subgroup $F(\mathbf{G})$ of the group $\mathbf{G}$ is the largest nilpotent subgroup in $\mathbf{G}$. Moreover by [1] the Fitting subgroup is formed by the left-Engel elements of the group $\mathbf{G}$. The following lemma shows that $\mathbf{G}/F(\mathbf{G})$ controls the properties of $\mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$:

Lemma 186. *Let* g_1, g_2 *be two arbitrary elements of* $\mathbf{G}$ *and let* $b_1 = \varphi(g_1)$*,* $b_2 = \varphi(g_2)$*. Then* $b_1 - b_2 \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$ *if and only if* $g_1 g_2^{-1} \in F(\mathbf{G})$*.*

Proof. Suppose first that $b_1 - b_2 \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$. Then $b_2^{-1}(b_1 - b_2) = b_1 b_2^{-1} - 1 \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$, thus $b_1 b_2^{-1} - 1$ is nilpotent in $\mathbf{R}(\mathbf{B})$. This means that commuting in $\mathbf{G}$ with the element $g_1 g_2^{-1}$ is a nilpotent action, i.e. $g_1 g_2^{-1}$ is a left-Engel element. The set of left-Engel elements form the fitting subgroup [1], hence $g_1 g_2^{-1} \in F(\mathbf{G})$.

Conversely, if $g_1 g_2^{-1} \in F(\mathbf{G})$, then commuting with $g_1 g_2^{-1}$ is a nilpotent action, i.e. $b_1 b_2^{-1} - 1 \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$. Then $\left(b_1 b_2^{-1} - 1\right) b_2 = b_1 - b_2 \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$, too. □

Let $\pi\colon \mathbf{R}(\mathbf{B}) \to \oplus_{i=1}^{k} \mathbf{F}_i = \mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$ the natural homomorphism. For every $1 \leq i \leq k$ let π_i be the projection from $\mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$ to $\mathbf{F}_i$. Now let

$$S = \{\, \pi(b + \mathbf{J}(\mathbf{R}(\mathbf{B}))) \mid b \in \mathbf{B} \,\},$$
$$S_i = \{\, \pi_i(b + \mathbf{J}(\mathbf{R}(\mathbf{B}))) \mid b \in \mathbf{B} \,\}.$$

Let $q_i = |\mathbf{S}_i|$ and let $q_0 = \max_{1 \leq i \leq k} q_i$. Let i_0 be an index for which $q_{i_0} = q_0$.

Lemma 187. *The following statements hold:*

1. $\pi \colon \mathbf{R}(\mathbf{B}) \to \oplus_{i=1}^{k} \mathbf{F}_i$ *is a ring-homomorphism.*
2. $\pi \colon \mathbf{B} \to S$ *is a group-homomorphism, which is an isomorphism between* $\mathbf{G}/F(\mathbf{G})$ *and* S.
3. $\mathbf{S}$ *is a multiplicative cyclic subgroup of* $\oplus_{i=1}^{k} \mathbf{F}_i$.
4. $\mathbf{S}_i$ *is a multiplicative cyclic subgroup of* $\mathbf{F}_i$.
5. $\mathbf{S}$ *generates the ring* $\oplus_{i=1}^{k} \mathbf{F}_i$.
6. $\mathbf{S}_i$ *generates the ring* $\mathbf{F}_i$.
7. $q_i \mid |\mathbf{F}_i| - 1$.
8. $q_i \mid q$.
9. *Let* $g \in \mathbf{G}$ *such that for some integer* m *we have* $g^m \in F(\mathbf{G})$ *and* $g^j \notin F(\mathbf{G})$ *for every* $1 \leq j \leq m-1$. *Then there exist* $1 \leq i \leq k$ *such that* $m \mid q_i$.
10. *If for a prime* p *we have* $p^\alpha \mid q$, *then there exists an* i *such that* $1 \leq i \leq k$ *and* $p^\alpha \mid q_i$.

Proof. Item 1 is by definition, item 2 is a consequence of Lemma 186. Item 3 and item 4 follows from item 2. Item 5 and item 6 can be derived from the fact that $\mathbf{B}$ generates $\mathbf{R}(\mathbf{B})$. Item 7 follows from item 3. For item 8 let $s \in S$. There exist an element $g \in \mathbf{G}$ and an element $b \in \mathbf{B}$ such that $\varphi(g) = b$ and $\pi(b) = s$. Now $g^q \in F(\mathbf{G})$, therefore by item 2 we have s^q is the identity element in $\oplus_{1 \leq i \leq k} \mathbf{F}_i$. This means $q_i \mid q$, which is item 8. For item 9 let $g \in \mathbf{G}$ and element such that $g^m \in F(\mathbf{G})$ and $g^j \notin F(\mathbf{G})$ for every $1 \leq j \leq m-1$. Let $b = \varphi(g)$ and let $s = (s_1, \ldots, s_k) = \pi(b)$. Since $g^j \notin F(\mathbf{G})$ for $1 \leq j \leq m-1$, and φ and π are homomorphisms, $b^j \notin \mathbf{J}(\mathbf{R}(\mathbf{B}))$ and $s^j \neq (1, \ldots, 1)$. However $g^m \in F(\mathbf{G})$, therefore $s^m = (1, \ldots, 1)$. This means that there is a coordinate i such that the order of s_i is exactly m, hence $m \mid q_i$. Finally for item 10 we use item 9 with $m = p^\alpha$. □

Lemma 188. *If* $\exp \mathbf{G}/F(\mathbf{G}) \geq 3$ *then* $\max_{1 \leq i \leq k} q_i = q_0 \geq 3$.

Proof. By item 10 from Lemma 187 we know that there exists q_i such that q_i is at least the largest prime power factor of q. Since $q \geq 3$, its largest prime power factor is at least 3. Therefore $q_0 \geq 3$. □

Remark 189. By Lemma 187 we have that q_0 is at most $\exp(\mathbf{G}/F(\mathbf{G}))$ and is at least the largest prime power divisor of $\exp(\mathbf{G}/F(\mathbf{G}))$. Both of these bounds are sharp, as the following two groups show:

$$\begin{aligned} \mathbf{G}_1 &= \langle a, b \mid a^7 = b^6 = 1, b^{-1}ab = a^3 \rangle && (q = q_0 = 6), \\ \mathbf{G}_2 &= \mathbf{S}_3 \oplus \mathbf{A}_4 && (q = 6, q_0 = 3). \end{aligned}$$

Now we continue on the proof of Theorem 184.

Proof of Theorem 184. In the proof of Theorem 174 we introduced the following operation:

$$f(y, \bar{x}_1, \bar{x}_2) = y^{p(\bar{x}_1) - p(\bar{x}_2)} = y^{z_1 - z_2},$$

using the notation $z_i = p(\bar{x}_i)$. However, if z_2 is invertible, then $y^{z_1 - z_2} = (y^{z_2})^{z_1 z_2^{-1} - 1}$, and if y runs through the elements of a normal subgroup, then y^{z_2} runs through the elements of the same normal subgroup. Moreover, if $z_1 = \varphi(g_1)$ and $z_2 = \varphi(g_2)$, then $y^{z_1 - z_2} = \left[y^{g_2}, g_1 g_2^{-1}\right]$. Using this idea we change f to the commutator of $\mathbf{G}$.

The proof consists of the following steps:

1. Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We exhibit a word u_Γ over $\mathbf{R}(\mathbf{B})$ such that $_\Gamma = 0$ in $\mathbf{R}(\mathbf{B})$ for every substitution of the variables from $\mathbf{B}$ if and only if Γ is *not* q_0-colorable (Lemmas 190, 191 and 193).

2. For every graph Γ we exhibit a word $Q_\Gamma = a^{u_\Gamma}$ over $(\mathbf{G}, [,])$ and Lemma 194 proves that $(\mathbf{G}, [,]) \models Q_\Gamma \approx 1$ if and only if Γ is *not* q_0-colorable.

3. We prove that the length of Q_Γ over $(\mathbf{G}, [,])$ is polynomial in the size of Γ. Thus we polynomially reduced the GRAPH q_0-COLORING problem (for some $q_0 > 2$) to the equivalence problem over $(\mathbf{G}, [,])$.

We start with step 1 of the proof. Let $\Gamma = (V, E)$ be an arbitrary simple graph with no loops, or multiple edges, $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Let u'_Γ and u_Γ be the following ring-expressions:

$$\begin{aligned} u'_\Gamma(x_1, \ldots, x_n) &= \prod_{v_i v_j \in E} (x_i x_j^{-1} - 1), \\ u_\Gamma(x_1, \ldots, x_n) = (u'_\Gamma(x_1, \ldots, x_n))^{e_0} &= \prod_{v_i v_j \in E} (x_i x_j^{-1} - 1)^{e_0}. \end{aligned}$$

Lemma 190. *For every $1 \leq i \leq k$ we have $u'_\Gamma = 0$ in $\mathbf{F}_i$ for every substitutions of the variables from S_i if and only if Γ is* not *q_i-colorable.*

Proof. We color the vertices of Γ by the elements of S_i. The color of v_j will be s_j. We prove that $u'_\Gamma(s_1, \ldots, s_n) \neq 0$ if and only if the appropriate coloring is a q_i-coloring of Γ.

First, let us assume that Γ is q_i-colorable, and let s_j be the color of v_j. Now, substituting $x_j = s_j$, for every edge $e = v_{j_1}v_{j_2}$ of Γ we have $x_{j_1}x_{j_2}^{-1} - 1 \neq 0$, hence $u'_\Gamma(s_1, \ldots, s_n) \neq 0$. Conversely, if Γ is not q_i-colorable, then at any assignment of the variables we have a monochromatic edge, $e = v_{j_1}v_{j_2}$. Then $u'_\Gamma = 0$ at every substitution from S_i. □

Lemma 191. *We have $u'_\Gamma = 0$ in $\oplus_{1\leq i\leq k}\mathbf{F}_i$ for every substitutions of the variables from S if and only if Γ is* not *q_0-colorable.*

Proof. If Γ is q_0-colorable, then by Lemma 190 there exists a substitution of the variables from S_{i_0} such that $u'_\Gamma \neq 0$ in $\mathbf{F}_{i_0}$. Let us extend this substitution to a substitution from S, then we have $u'_\Gamma \neq 0$ in $\oplus\mathbf{F}_i$ for this substitution. If Γ is not q_0-colorable, then it is not q_i-colorable for any $1 \leq i \leq k$, thus $u'_\Gamma = 0$ for every substitution from S_i (for every $1 \leq i \leq k$). Hence $u'_\Gamma = 0$ in $\oplus\mathbf{F}_i$ for every substitutions from $\oplus S_i$, and so from S. □

Remark 192. We note here that $S \leq \oplus S_i$, but they are not necessarily equal as the following example shows:

$$\begin{aligned}\mathbf{G} &= \left\langle a, b, c \mid a^5 = b^5 = c^4 = 1, b^{-1}ab = a, c^{-1}ac = a^2, c^{-1}bc = b^3 \right\rangle \\ &\simeq (\mathbf{Z}_5 \oplus \mathbf{Z}_5) \rtimes \mathbf{Z}_4.\end{aligned}$$

Lemma 193. *We have $u_\Gamma = 0$ in $\mathbf{R}(\mathbf{B})$ for every substitutions of the variables from $\mathbf{B}$ if and only if Γ is* not *q_0-colorable.*

Proof. By $u_\Gamma = (u'_\Gamma)^{e_0}$, we have $u_\Gamma = 0$ for some substitution from $\mathbf{B}$ if and only if $u'_\Gamma \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$ for the same substitution.

If Γ is q_0-colorable, then by Lemma 191 there exists a substitution from S such that $u'_\Gamma \neq 0$ in $\oplus\mathbf{F}_i = \mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$. This substitution has a pre-image in $\mathbf{B}$, and for the pre-image substitution we have $u'_\Gamma \notin \mathbf{J}(\mathbf{R}(\mathbf{B}))$.

If Γ is not q_0-colorable, then by Lemma 191 for every substitution from S we have $u'_\Gamma = 0$ in $\oplus\mathbf{F}_i = \mathbf{R}(\mathbf{B})/\mathbf{J}(\mathbf{R}(\mathbf{B}))$. This means that for every substitution from $\mathbf{B}$ we have $u'_\Gamma \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$. □

We continue with step 2 of the proof. Now, we polynomially reduce the GRAPH q_0-COLORING problem to the equivalence problem over $(\mathbf{G}, [,])$. Let $\Gamma = (V, E)$ be a graph, with n vertices, $V = \{v_1, \ldots v_n\}$ and m edges,

$E = \{e_1, \dots e_m\}$. Let $x_1, \dots, x_n$ be different variables assigned to the vertices. Let us denote all these 'x' variables by $\bar{x} = (x_1, \dots, x_n)$. For every edge e_i let $e_i = v_{i,1}v_{i,2}$ and let

$$\begin{aligned} w_1(y, \bar{x}) &= \left[y, x_{1,1}x_{1,2}^{-1}\right] = y^{\left(x_{1,1}x_{1,2}^{-1}-1\right)}, \\ w_i(y, \bar{x}) &= \left[w_{i-1}(y, \bar{x}), x_{i,1}x_{i,2}^{-1}\right] \\ &= y^{\left(x_{1,1}x_{1,2}^{-1}-1\right)\dots\left(x_{i,1}x_{i,2}^{-1}-1\right)}, \end{aligned}$$

where $x_{i,j}$ is the variable assigned to the vertex $v_{i,j}$. Observe, that $w_m = y^{u'_\Gamma(z_1,\dots,z_n)}$ with the notation $z_i = p(\bar{x}_i)$. Let

$$\begin{aligned} W_1(y, \bar{x}) &= w_m(y, \bar{x}) = y^{u'_\Gamma(z_1,\dots,z_n)}, \\ W_i(y, \bar{x}) &= W_1 \circ W_{i-1} = W_1(W_{i-1}(y, \bar{x}), \bar{x}) \\ &= y^{\left(u'_\Gamma(z_1,\dots,z_n)\right)^i}. \end{aligned}$$

Now $\mathbf{A} = \mathbf{N}$ is a verbal subgroup of $\mathbf{G}$, let $W_0(\bar{y})$ be a word with image $\mathbf{A}$. We are interested in $Q_\Gamma = W_{e_0}(W_0(\bar{y}), x_{1,1}, x_{1,2}, \dots, x_{m,1}, x_{m,2})$, where e_0 was the natural number for which $(r^{e_0})^2 = r^{e_0}$ for every $r \in \mathbf{R}(\mathbf{B})$ and $r^{e_0} = 0$ if and only if $r \in \mathbf{J}(\mathbf{R}(\mathbf{B}))$. Observe, that $Q_\Gamma = W_0(\bar{y})^{u_\Gamma(x_1,\dots,x_n)}$.

Lemma 194. *The graph Γ is q_0-colorable if and only if $(\mathbf{G}, [,]) \models Q_\Gamma \not\approx 1$.*

Proof. If Γ is q_0-colorable, then by Lemma 193 there exists a substitution of $x_1, \dots, x_n$ from $\mathbf{B}$ such that $u_\Gamma(x_1, \dots, x_n) \neq 0$ in $\mathbf{R}(\mathbf{B})$. Thus there exists an $a \in \mathbf{A}$ such that $a^{u_\Gamma} \neq 1$ in $\mathbf{G}$ for the same substitution. Choose $\bar{y}$ such that $a = W_0(\bar{y})$ and with this evaluation of the variables we have that $(\mathbf{G}, [,]) \models Q_\Gamma \not\approx 1$. If Γ is not q_0-colorable, then we have that $u_\Gamma \approx 0$ in $\mathbf{R}(\mathbf{B})$ for every substitution of the variables from $\mathbf{B}$. Thus for every $a \in \mathbf{A}$ (especially $a = W_0(\bar{y})$) we have $a^{u_\Gamma} = 1$ and $(\mathbf{G}, [,]) \models Q_\Gamma \approx 1$. □

We finish with step 3 of the proof. Denote the length of an expression w with $\|w\|$. The reduction from GRAPH q_0-COLORING to the equivalence problem over $(\mathbf{G}, [,])$ is polynomial, because the length of $Q_\Gamma = Q_{e_0}$ is $\|Q_\Gamma\| = \|Q_{e_0}\| = O(m \cdot e_0 \cdot (n + \|S_0\|))$: when building up Q_Γ we use the commutator exactly $e_0 \cdot m$-many times on first input of length $\|W_0\|$ and on n-many variables. Hence $\|Q_\Gamma\|$ is polynomial in the size of Γ and Theorem 184 is proved. □

8.6 Problems

In Section 8.5 we did not consider any non-nilpotent groups $\mathbf{G}$ for which both $\mathbf{N} = \mathbf{N}(\mathbf{G})$ (defined in Section 8.2) and $C_{\mathbf{G}}(\mathbf{N})$ are Abelian and

$\exp \mathbf{G}/F(\mathbf{G}) = 2$. Checking the proof of Theorem 169 we observe that if the complexity of the equivalence problem for $(\mathbf{G}, [,])$ is coNP-complete for such groups, then Theorem 169 would follow by induction with f being the commutator of the group. If, however, this is not the case, then the characterization would be much harder:

Problem 9. Characterize those non-nilpotent finite groups $\mathbf{G}$, for which the equivalence problem for $(\mathbf{G}, [,])$ is coNP-complete!

The first step on the way answering Problem 9 would be to check the smallest possible group for which we do not know this complexity.

Problem 10. What is the complexity of the equivalence problem for $(\mathbf{S}_3, [,])$?

Chapter 9

Summary and next directions

In the thesis we investigated the relationship of functions and their realizing polynomials over finite algebras. We studied functionally complete algebras, i.e. algebras over which every function can be realized by a polynomial expression. In Chapter 2 we characterized functionally completeness by the Stone–Weierstrass property. While the functionally complete rings and functionally complete groups are all described, we determined the functionally complete semigroups in Section 2.4 and the functionally complete semirings in Section 2.5.

From Chapter 3 we were especially interested about the computational perspective of the function–polynomial relationships over finite groups. We considered three themes regarding polynomials over algebras.

1. The efficient representability problem.

2. The equivalence problem.

3. The equation solvability problem.

We approached the efficient representability problem from three directions. We considered the length of functions in Chapter 3. We investigated the circuit complexity of functions in Sections 4.1, 4.2, 4.3 and 4.4. Finally we analysed the finite-state sequential machine representation of Boolean functions in Section 4.5. We observed that computers based on functionally complete groups do not seem to be more efficient than the usual two-element Boolean algebra based computers in general, but they might be more efficient in special circumstances. Finding several examples of functions which can be represented more efficiently by functionally complete groups could be a next step of this research.

Neither the equivalence problem nor the equation solvability problem has been completely characterized for finite groups. The complexity of the equivalence problem is known for nilpotent groups, and we determined the complexity for non-solvable groups in Chapter 7. Not much is known about the case of solvable, non-nilpotent groups: we provide results for some meta-Abelian groups in Section 6.1. It is likely that, with a deeper investigation of solvable, non-nilpotent groups, the characterization of the equivalence problem for finite groups can be finished.

The complexity of the equation solvability problem is known for nilpotent groups and for non-solvable groups. There are no results about the complexity of the equation solvability problem for solvable, non-nilpotent groups apart from the case of certain meta-cyclic groups that we presented in Section 6.2.

The idea of the extended equivalence problem emerged from an observation of Section 3.6, namely that the commutator might significantly change the length of group-polynomials. In Chapter 8 we characterized the complexity of the extended equivalence problem for finite groups. For many finite groups $\mathbf{G}$ we determined the complexity of the equivalence problem for $(\mathbf{G}, [,])$, but a complete characterization is still required.

Bibliography

[1] R. Baer. Engelsche elemente noetherscher gruppen. *Math. Ann.*, 133:256–270, 1957.

[2] S. Burris and J. Lawrence. The equivalence problem for finite rings. *Journal of Symbolic Computation*, 15:67–71, 1993.

[3] S. Burris and J. Lawrence. Results on the equivalence problem for finite groups. *Algebra Universalis*, 52(4):495–500, 2004. (2005).

[4] P. J. Cameron. *Permutation Groups*. LMS Student Text. Cambridge University Press, Cambridge, 1999.

[5] A. H. Clifford and G. B. Preston. *The Algebraic Theory of Semigroups*, volume 1. American Mathematical Society, 1961.

[6] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Undergraduate Texts in Mathematics. Springer, 2nd edition, 1996.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., San Francisco, 1979.

[8] S. B. Gaskov. The depth of Boolean functions. *Problemy Kibernetiki*, 34:265–268, 1978. in Russian.

[9] K. Głazek. *A Guide to the Literature on Semirings and Their Applications in Mathematics and Information Sciences*. Kluwer Academic Publishers, Dordrecht, 2002. With complete bibliography.

[10] M. Goldmann and A. Russell. The complexity of solving equations over finite groups. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 80–86, Atlanta, Georgia, May 1999.

[11] U. Hebisch and H. J. Weinert. *Semirings: Algebraic Theory and Applications in Computer Science*, volume 5 of *Series in Algebra*. World Scientific Publishing Co. Inc., River Edge, NJ, 1998. Translated from the 1993 German original.

[12] J. L. Henessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, fourth edition, 2006.

[13] G. Horváth, J. Lawrence, L. Mérai, and Cs. Szabó. The complexity of the equivalence problem for non-solvable groups. *Bulletin of the London Mathematical Society*, 39(3):433–438, 2007. doi:10.1112/blms/bdm030.

[14] G. Horváth, C. L. Nehaniv, and Cs. Szabó. An assertion concerning functionally complete algebras and NP-completeness. *Theoretical Computer Science*, 2007. submitted.

[15] G. Horváth and Cs. Szabó. The complexity of checking identities over finite groups. *International Journal of Algebra Computation*, 16(5):931–940, October 2006.

[16] H. Hunt and R. Stearns. The complexity for equivalence for commutative rings. *Journal of Symbolic Computation*, 10:411–436, 1990.

[17] N. Jacobson. The radical and semi-simplicity for arbitrary rings. *Amer. J. Math.*, 67:300–320, 1945.

[18] M. Kilp, U. Knauer, and A. V. Mikhalev. *Monoids, Acts and Categories*, volume 29 of *de Gruyter Expositions in Mathematics*. Walter de Gruyter, Berlin, 2000.

[19] A. Kisielewicz. Complexity of semigroup identity checking. *International Journal of Algebra and Computation*, 14(4):455–464, 2004.

[20] O. Klíma. Complexity issues of checking identities in finite monoids. manuscript, http://math.muni.cz/∼klima/Math/coNPidcheck.ps.

[21] O. Klíma. *Unification Modulo Associativity and Idempotency*. PhD thesis, Masarik University, Brno, Czech Republic, 2004.

[22] K. Krohn, W. D. Maurer, and J. Rhodes. Realizing complex boolean functions with simple groups. *Information and Control*, 9(2):190–195, 1966.

[23] B. Larose and L. Zádori. Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. *International Journal of Algebra and Computation*, 16(3):563–581, 2006.

[24] O. B. Lupanov. On a method of circuit synthesis. *Izv. VUZ, Radiofizika*, 1:120–140, 1958. in Russian.

[25] O. B. Lupanov. Complexity of formula realization of functions of logical algebra. *Probl. Cybernetics*, 3:782–811, 1962.

[26] W. D. Maurer and J. L. Rhodes. A property of finite simple non-abelian groups. *Proc. Amer. Math. Soci.*, 16:552–554, 1965.

[27] R. N. McKenzie, G. F. McNulty, and W. F. Taylor. *Algebras, Lattices, Varieties.*, volume 1. The Wadsworth & Brooks/Cole Mathematics Series. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, Calif., 1987. ISBN: 0-534-07651-3.

[28] H. Neumann. *Varieties of Groups*. Springer-Verlag, Berlin, 1967.

[29] T. Nipkow. Unification in primal algebras, their powers and their varieties. *Journal of the Association for Computing Machinery*, 37(1):742–776, October 1990.

[30] E. L. Post. Introduction to a general theory of elementary propositions. *Amer. J. Math.*, 43:163–185, 1921.

[31] D. J. S. Robinson. *A Course in the Theory of Groups*. Springer-Verlag, New York, Berlin, Heidelberg, 1995.

[32] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Science/Engineering/Math, 3rd edition, 1976.

[33] J. E. Savage. *The Complexity of Computing*. John Wiley and Sons Inc., New York, 1976.

[34] S. Seif and Cs. Szabó. Computational complexity of checking identities in 0-simple semigroups and matrix semigroups over finite fields. *Semigroup Forum*, 72(2):207–222, 2006.

[35] P. M. Spira. The time required for group multiplication. *Journal of Association for Computing and Machinery*, 16(2):235–243, April 1969.

[36] M. H. Stone. The representation of boolean algebras. *Bull. Amer. Math. Soc.*, 44:807–816, 1938.

[37] P. Tesson. *Computational Complexity Questions Related to Finite Monoids and Semigroups.* PhD thesis, McGill University, Montreal, 2004.

[38] P. Tesson and D. Thérien. Monoids and computations. *International Journal of Algebra and Computation*, 14(5-6):801–816, 2004.

[39] M. V. Volkov. Checking identities in semigroups. Lecture presented at the Conference on Universal Algebra, Nashville, 2002.

[40] I. Wegener. *The Complexity of Boolean Functions.* John Wiley & Sons Ltd, and B. G. Teubner, Stuttgart, 1987.

[41] H. Werner. *Einführung in die Allgemeine Algebra.* Bibliographisches Institut, Mannheim/Wien/Zürich, 1978.

[42] J. S. Wilson. Finite axiomatization of finite soluble groups. *J. London Math. Soc.*, 74(3):566–582, 2006.

[43] S. Winograd. On the time required to perform addition. *Journal of Association for Computing and Machinery*, 12(2):277–285, April 1965.

[44] S. Winograd. On the time required to perform multiplication. *Journal of Association for Computing and Machinery*, 14(4):793–802, October 1967.

Appendix A

Statement on joint work

Chapter 7 is published as a joint paper with not only my secondary supervisor Csaba Szabó, but with László Mérai and John Lawrence. My contribution to this joint work was Lemma 165 and the final reduction from a non-solvable group to a simple group.

The results of other Chapters are mine, unless explicitly indicated otherwise.